OPTIMAL SHAPE DESIGN BASED ON BODY-FITTED GRID GENERATION

by

Farzad Mohebbi

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering at the University of Canterbury, Christchurch, New Zealand

Supervisor: Dr Mathieu Sellier

July 2014

Dedication

To my lovely wife, Rahil

Acknowledgments

I would like to express my sincere thanks to my supervisor, Dr Mathieu Sellier, for giving me this great opportunity to do PhD and also for his invaluable and endless advice, support, and encouragement during the preparation of this thesis. I have benefited by his insightful comments and very helpful suggestions throughout the course of this research work.

The partial financial support of the Department of Mechanical Engineering at the University of Canterbury through the Premier Departmental Scholarship is gratefully appreciated.

I have been most fortune in my PhD student career to have had ample opportunity to act as a teaching assistant for various courses in undergraduate level in Mechanical Engineering, Civil Engineering, and Mathematics. I have been afforded a great deal of experience with a wide range of courses including Statics, Dynamics, Mechanics of Materials, Heat Transfer, Fluid Mechanics, Engineering Mathematics, and Finite Element Analysis. I wish to express my deepest thanks to the academic staff in the departments of Mechanical Engineering, Civil Engineering, and Mathematics for giving me this opportunity to work as a Teaching Assistant for them.

I would like to express my sincere gratitude to my lovely wife, Rahil, for all her endless love and support in entire my life with her. Without her support, finishing this PhD would not have been possible.

Last but not least, my deepest thanks go to my parents, sister, and brothers for all their endless love, support, and kindness during all my study.

Abstract

Shape optimization is an important step in many design processes. With the growing use of Computer Aided Engineering in the design chain, it has become very important to develop robust and efficient shape optimization algorithms. The field of Computer Aided Optimal Shape Design has grown substantially over the recent past.

In the early days of its development, the method based on small shape perturbation to probe the parameter space and identify an optimal shape was routinely used. This method is nothing but an educated trial and error method. A key development in the pursuit of good shape optimization algorithms has been the advent of the adjoint method to compute the shape sensitivities more formally and efficiently. While undoubtedly, very attractive, this method relies on very sophisticated and advanced mathematical tools which are an impediment to its wider use in the engineering community. It that spirit, it is the purpose of this thesis to propose a new shape optimization algorithm based on more intuitive engineering principles and numerical procedures.

In this thesis, the new shape optimization procedure which is proposed is based on the generation of a body-fitted mesh. This process maps the physical domain into a regular computational domain. Based on simple arguments relating to the use of the chain rule in the mapped domain, it is shown that an explicit expression for the shape sensitivity can be derived. This enables the computation of the shape sensitivity in one single solve, a performance analogous to the adjoint method, the current state-of-the art. The discretization is based on the Finite Difference method, a method chosen for its simplicity and ease of implementation.

This algorithm is applied to the Laplace equation in the context of heat transfer problems and potential flows. The applicability of the proposed algorithm is demonstrated on a number of benchmark problems which clearly confirm the validity of the sensitivity analysis, the most important aspect of any shape optimization problem. This thesis also explores the relative merits of different minimization algorithms and proposes a technique to “fix” meshes when inverted element arises as part of the optimization process.

While the problems treated are still elementary when compared to complex multiphysics engineering problems, the new methodology presented in this thesis could apply in principle to arbitrary Partial Differential Equations.

Publications

During the course of this research work, the following publications are appeared:

Journal articles:

1. **Mohebbi, F.** and Sellier, M. (2013) Optimal shape design in heat transfer based on body - fitted grid generation. *International Journal for Computational Methods in Engineering Science and Mechanics*, vol. 14, issue 3, pp. 227-243 (http://dx.doi.org/10.1080/15502287.2012.711426)

2. **Mohebbi, F.** and Sellier, M. (2013) Three - dimensional optimal shape design in heat transfer based on body - fitted grid generation. *International Journal for Computational Methods in Engineering Science and Mechanics* (http://dx.doi.org//10.1080/15502287.2013.784384)

3. Mohebbi, F. and Sellier, M. (2014) On the Kutta condition in potential flow over airfoil. *Journal of Aerodynamics*

(http://dx.doi.org/10.1155/2014/676912)

4. **Mohebbi, F.** and Sellier, M. (2014) Aerodynamic Optimal shape design based on body - fitted grid generation. *Mathematical Problems in Engineering*

(http://www.hindawi.com/journals/mpe/aip/505372)

Contents

Part one: Shape Optimization in Heat Transfer

List of Tables

Nomenclature

$\mathbf{d}^{(\mathrm{k})}$	direction of descent at iteration k
$\dot{q}$	heat flux $(\frac{\mathrm{W}}{\mathrm{m}^2})$
h	heat transfer coefficient $(\frac{\mathrm{W}}{\mathrm{m}^2.^\circ\mathrm{C}})$
$\mathbf{Ja}$	Jacobian matrix
J	Jacobian of transformation
k_T	thermal conductivity of the solid body $(\frac{\mathrm{W}}{\mathrm{m}.^\circ\mathrm{C}})$
$\mathbf{n}$	outward drawn unit vector
$\boldsymbol{\mathcal{J}}$	objective function
P, Q, R	grid control functions
S	boundary surface
t	time
T	temperature ($^\circ$C)
T_d	desired outer surface temperature ($^\circ$C)
T_∞	ambient temperature ($^\circ$C)
x, y, z	Cartesian coordinates in the physical domain

Greek symbols

α, β, γ	metric coefficients in 2-D elliptic grid generation
$\alpha_{11}, \ldots, \alpha_{33}$	metric coefficients in 3-D elliptic grid generation
$\beta^{(\mathrm{k})}$	search step size at iteration k
$\varepsilon_1, \varepsilon_2$	stopping criteria predefined values
Γ	boundary
$\gamma^{(\mathrm{k})}$	conjugation coefficient at iteration k
Ω	domain
ξ, η, ζ	Cartesian coordinates in the computational domain

Subscripts

i	grid index in ξ - direction
j	grid index in η - direction
k	grid index in ζ - direction
M	number of grid points in the ξ - direction
N	number of grid points in the η - direction
L	number of grid points in the ζ - direction

Superscript

k	iteration number

Abbreviations

OSD	Optimal Shape Design
PDE	Partial Differential Equation
FEM	Finite Element Method
FDM	Finite Difference Method
LMA	Levenberg-Marquardt Algorithm
CGM	Conjugate Gradient Method
TFI	TransFinite Interpolation
TR	Tikhonov's Regularization
SOR	Successive Over Relaxation

Part one:

Shape Optimization in Heat Transfer

1. Introduction

1.1 Inverse heat transfer problem

Traces of Inverse Problems can be found in almost every branch of science and engineering such as mechanical, structural, chemical, aerospace, physics, and mathematics. Inverse Heat Transfer Problems (IHTP) encompass a broad range of inverse problems which have received much attention from both practical and theoretical viewpoints and have been greatly researched by engineers and mathematicians during the last decades. Therefore, studying the procedures to solve such complicated problems including the numerical implementation and improving the efficiency of employed methods is of crucial importance.

The study of IHTP dates back to the beginning of the space program in 1957. The space program gave significant impetus to the study of IHTP [1]. On reentrance into the atmosphere, direct measurement of the high temperature at the surface of the thermal shields of space vehicles with temperature sensors was impossible. Hence, the sensors were placed at a specified distance from the surface and an inverse analysis was employed to calculate the hot temperature at the surface.

One of the earliest papers on IHTP, which was published by Stolz [2] in 1960, dealt with the calculation of heat transfer rates during quenching of the bodies of simple finite shapes. He claimed the use of his method in 1957. Mirsepassi stated that he had used the same method both numerically and graphically for semi-infinite geometries prior to 1960 [3, 4]. A Russian paper by Shumakov on the IHTP was translated in 1957 [5]. Giedt examined the heat transfer at the inner surface of a gun barrel [6].

1.2 Definition

If all information on the boundary conditions, the thermo-physical properties, the geometrical configuration of heated body, and the heat flux of a heat transfer problem is known but the temperature distribution is unknown, the

heat transfer problem is referred to as the *direct heat transfer problem*. If either of this information is unknown, the heat transfer problem is referred to as the *inverse heat transfer problem.* As an illustrative example, the most common IHTP consists in recovering the unknown heat flux in a conducting solid body subject to transient heat conduction by measuring the temperatures at different places and times [7-9] (see Figure 1-1). In this figure, x is the location of the sensor(s), $T_{measured}(x,t)$ is the measured temperature at location x and time t and right-hand side of the body is exposed to convection heat transfer.

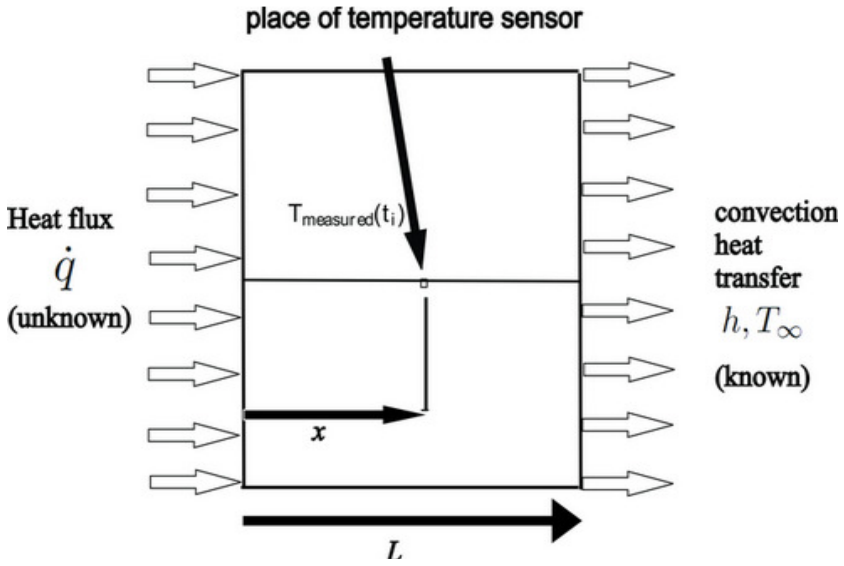


Figure 1-1 An inverse heat conduction problem. The known are the measured temperatures at location x (effect) and the unknown are the heat flux $\dot{q}$ on left hand side of the body (cause).

Applications of IHTP include the determination of thermo-physical properties such as the thermal conductivity and the heat transfer coefficient [10-17], the estimation of initial conditions such as the initial temperature [18, 19], the estimation of heat flux [7, 8, 20-22], and the estimation of the boundary shapes of bodies [23, 24], to name a few.

As mentioned above, one of the applications of IHTP consists in finding the optimal geometrical configuration of a conducting body with given thermo-physical properties and subject to given boundary conditions to minimize a certain objective function (see Figure 1-2) [25]. This objective function may be, for example, the total heat flux over a small subset of the body boundary or the difference between a computed temperature

distribution and a desired one. This field is known as *shape optimization* or *optimal shape design* (OSD). Shape optimization in heat transfer problems will be further discussed subsequently.

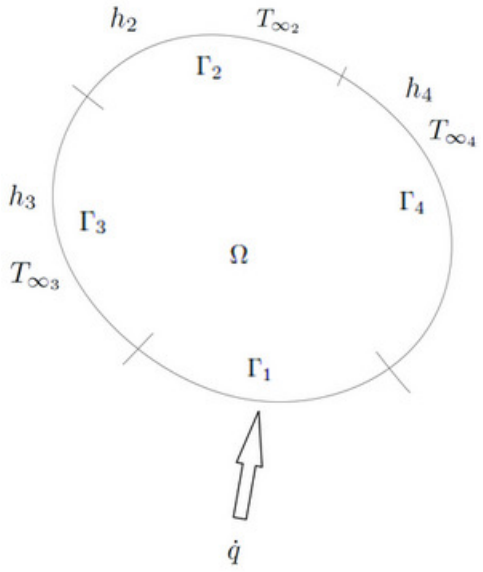


Figure 1-2 An inverse heat transfer problem. The known are the convective heat transfer coefficients $h_i (i = 2,3,4)$, ambient temperatures $T_{\infty_i} (i = 2,3,4)$, thermal conductivity of the body k_T, and heat flux $\dot{q}$. The unknown is the shape of the heated body.

1.3 Ill-posedness

IHTPs have inherent difficulties which should first be introduced. This class of problems is *ill-posed* whereas direct heat transfer problems are *well-posed* [1, 26-28]. The difficulty of solving ill-posed problems results from the fact that they are inherently unstable and very sensitive to noise and perturbation. For considering a problem as well-posed, three conditions must be satisfied [9].

- a solution must exist
- this solution must be unique
- the solution must be stable with respect to perturbation of the input data

Hadamard (1923) classified the inverse problems as ill-posed as the three conditions are not necessarily satisfied [29]. In general, to solve an

inverse problem, one needs to reformulate the inverse problem as an approximate well-posed one by using some stabilization (regularization) techniques [25]. Three important methods to overcome the instabilities in inverse heat transfer problems have been proposed. The first one by Tikhonov (*Tikhonov's regularization procedure*) [26, 30], the second one by Alifanov (*Alifanov's iterative regularization techniques*) [27, 31, 32], and the last one by Beck (*Beck's function estimation approach*) [1, 7, 28].

The objective of IHTP shown in Figure 1-1 is to determine the heat flux $\dot{q}$ using the measured temperature history at location x and time t. In this problem, if $\mathbf{T}_e$ and $\mathbf{T}_m$ stand for vectors of estimated and measured temperatures, respectively, the objective function can be given by the *least square norm* as follows [28, 33-36]

$$\mathcal{J} = (\mathbf{T}_m - \mathbf{T}_e)^{\mathrm{T}}(\mathbf{T}_m - \mathbf{T}_e) \tag{1.1}$$

where superscript T stands for transpose of matrix. By considering the inverse heat transfer problem shown in Figure 1-1 and assigning the estimated values to unknown heat flux, the estimated temperatures are found by solving the direct heat transfer problem. If a single sensor is used for measuring the temperature at an interior point at times t_I, $I = 1,2,\ldots,M$ (see Figure 1-1), the objective function can be written as least squares norm in the form

$$\mathcal{J} = \left(\mathbf{T}_m - \mathbf{T}_e\right)^{\mathrm{T}}\left(\mathbf{T}_m - \mathbf{T}_e\right) = \sum_{I=1}^{M}(T_{m_I} - T_{e_I})^2 \tag{1.2}$$

Indeed, to solve an IHTP, one should reduce the mismatch between the estimated temperatures (obtained from the solution of the direct heat transfer problem using the estimated heat flux) and the measured temperature as much as possible. One method to reduce this mismatch consists in minimizing the least squares norm (Equation (1.2)) with respect to unknown quantities (the heat flux in this example). The minimization of the objective function given by Equation (1.2) may be stable if it involves the calculation of few unknown quantities. However, if a large number of unknown parameters is involved, such as the parameters describing the shape of a body in optimal shape design (the problem of parameter estimation), the solution of the above minimization problem can become unstable and oscillations occur during the iterative solution procedure. A possible successful method to overcome these instabilities is the use of the *Tikhonov's regularization procedure*.

1.4 Tikhonov's Regularization (TR) procedure

This method is named for Russian Mathematician, Andrey Tikhonov, and is most commonly used method in the regularization of ill-posed inverse problems. This method modifies the least square norms by adding a penalty term to Equation (1.2) of the form $\alpha\sum_{I=1}^{M} f_I^2$, that is,

$$\mathcal{J} = \sum_{I=1}^{M}(T_{m_I} - T_{e_I})^2 + \alpha\sum_{I=1}^{M} f_I^2 \tag{1.3}$$

or $$\mathcal{J} = \| T_{m_I} - T_{e_I} \|^2 + \alpha \| f_I \|^2$$

where α is the *regularization parameter* (a positive quantity) and $\|\cdot\|$ is the Euclidean norm [1]. For the problem shown in Figure 1-1, f_I is heat flux $\dot{q}$ at time t_I. The precise selection of α plays a significant role in reducing the instabilities of the inverse problem solution. There are different methods to determine the optimal value of α, such as the Morozov's *discrepancy principle* [37], the cross-validation approach, and the L-curve method [38, 39]. If $\alpha \to 0$, $\sum_{I=1}^{M} f_I^2$ will be large and the difference between estimated and measured temperatures approaches zero. Hence, the solution may involve oscillations and becomes unstable. On the other hand, if we take large values of α, the solution may be damped and diverts from the exact solution. By differentiating Equation (1.3) with respect to unknown heat flux vector, equating the result with zero, and then analyzing the sensitivity coefficients, the following relationship is obtained

$$\dot{\mathbf{q}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \alpha\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{T}_m \tag{1.4}$$

where

---

[1] For a vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, $\| \mathbf{x} \| = \sqrt{(x_1^{\,2} + x_2^{\,2} + \ldots + x_n^{\,2})}$.

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \vdots \\ \dot{q}_M \end{bmatrix}, \mathbf{T}_e = \begin{bmatrix} T_{e_1} \\ T_{e_2} \\ T_{e_3} \\ \vdots \\ T_{e_M} \end{bmatrix}, \mathbf{T}_m = \begin{bmatrix} T_{m_1} \\ T_{m_2} \\ T_{m_3} \\ \vdots \\ T_{m_M} \end{bmatrix}, \mathbf{X} \equiv \frac{\partial \mathbf{T}_e}{\partial \dot{\mathbf{q}}^{\mathrm{T}}} = \begin{bmatrix} \frac{\partial T_{e_1}}{\partial \dot{q}_1} & \frac{\partial T_{e_1}}{\partial \dot{q}_2} & \cdots & \frac{\partial T_{e_1}}{\partial \dot{q}_M} \\ \frac{\partial T_{e_2}}{\partial \dot{q}_1} & \frac{\partial T_{e_2}}{\partial \dot{q}_2} & \cdots & \frac{\partial T_{e_2}}{\partial \dot{q}_M} \\ & \vdots & & \\ \frac{\partial T_{e_M}}{\partial \dot{q}_1} & \frac{\partial T_{e_M}}{\partial \dot{q}_2} & \cdots & \frac{\partial T_{e_M}}{\partial \dot{q}_M} \end{bmatrix} \quad (1.5)$$

In the above relations, $\mathbf{X}$ is the sensitivity matrix. By utilizing an iterative optimization method, one can solve the Equation (1.4) and obtain the unknown heat flux vector $\dot{\mathbf{q}}$ [40]. In following chapters, the sensitivity matrix (Jacobian matrix) will be denoted by $\mathbf{Ja}$.

1.5 Alifanov's iterative regularization techniques

An alternative to Tikhonov's regularization method is to use the Alifanov's iterative regularization one. In contrast to Tikhonov's regularization method, there is no need to modify the original objective function in iterative regularization methods. In these gradient based methods, the number of iteration acts as the regularization parameter in Tikhonov's method and the termination of the iterations is carried out based on the discrepancy principle [25, 27, 32, 41, 42]. The discrepancy principle will be introduced in the following section. Two successful gradient methods employed in these methods are the *Steepest Descent* and *Conjugate Gradient* methods.

1.6 Stopping criterion for iterative regularization methods

If no measurement errors are involved, the following criterion may be used to terminate the iterations

$$\mathcal{J}^{(\mathrm{k})} < \varepsilon \tag{1.6}$$

where k and ε are iteration number and a sufficiently small number, respectively. However, in real world problems, there always exist errors in the measurement of the quantities (e.g. temperature) needed in the IHTP. In such situations, the discrepancy principle is used as a criterion to terminate the iteration.

The value required for stopping the iteration based on the discrepancy principle is obtained from

$$\left|T_m - T_e\right| \approx \sigma(\text{considered constant}) \tag{1.7}$$

$$\varepsilon^* = I\sigma^2 \tag{1.8}$$

where σ is the standard deviation of the measurement error which is assumed constant, and I is the number of transient measurements taken for each sensor. Therefore, the stopping criterion for this case is considered as

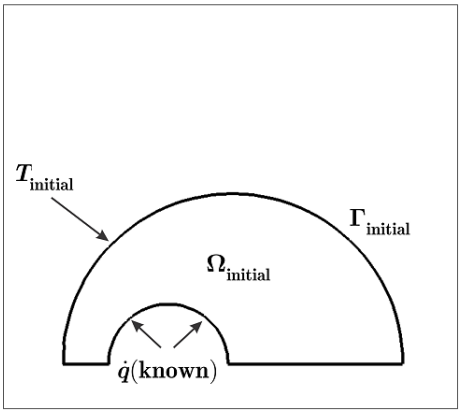
$$\mathcal{J}^{(\mathrm{k})} < \varepsilon^* \tag{1.9}$$

1.7 Shape optimization

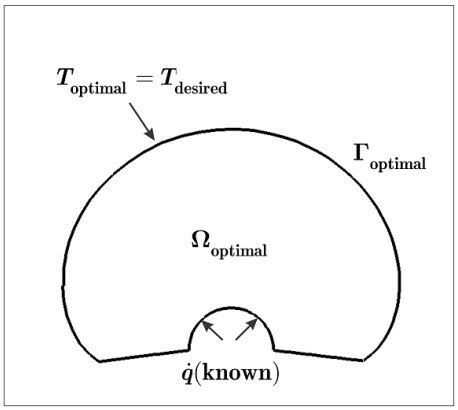
There are many applications for the shape optimization in heat transfer. Two applications are briefly stated here to state the importance of the shape optimization in heat transfer: Chemical Vapor Deposition and electronic packaging problems. Chemical Vapor Deposition (CVD) is a chemical process utilized to produce fine solid materials used in, say, semiconductor industry to produce thin films. Most important function of CVD heating system is to produce uniform temperature distribution across the surface of the substrate. Many factors involved in producing a fine material in CVD are strongly affected by the distribution of temperature throughout the CVD reactor. Therefore, the optimal shape design for the CVD heating system to produce a uniform temperature in the deposition zone and across the substrate surface is of vital importance. In electronic devices, the geometrical configuration of the conducting body containing internal heating elements plays an essential role in cooling performance (by dissipating the heat to the ambient by convective heat transfer) to prevent the accumulation of the heat in the device.

Therefore, an unsatisfactory shape design for such devices may cause damage to them.

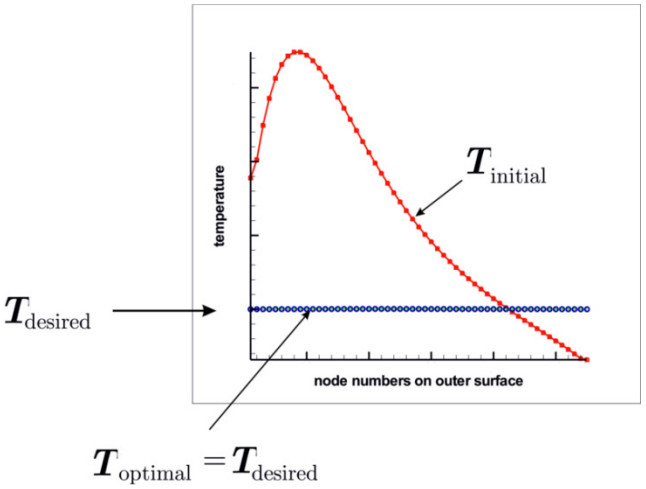
As stated previously, shape optimization in heat transfer may be considered as an inverse heat transfer problem whose aim is to recover the boundaries or shape of a heated body under specified boundary conditions. The steady state heat transfer problem of interest here consists in finding the optimal shape a body should have such that the temperature distribution on its boundary matches a desired distribution. The body is subject to a prescribed heat flux on part of its boundary and convective/radiative heat transfer on all other boundaries (see Figure 1-3).



a) The initially guessed shape.



b) The numerically obtained optimal shape.



c) The initial, optimal, and desired temperatures.

Figure 1-3 The initially guessed (a) and the numerically inferred optimal shapes (b). The objective is to match the initial and desired temperature distributions on the boundaries (c).

For the 3-D problem of interest here, the aim of shape optimization problem is to find the optimal shape for a body which is subject to a prescribed heat flux on part of its boundary, for example on Γ_5 ($ABFE$ in Figure 1-4), and convective/radiative heat transfer on all other boundaries $\Gamma_i (i = 1,2,3,4,6)$ (Figure 1-4).

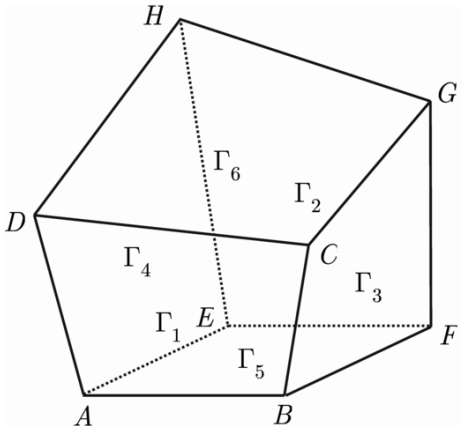


Figure 1-4 arbitrarily shaped heat conducting body under specified boundary conditions.

To deal with the shape optimization problems in heat transfer, attention is usually given to the use of a combination of the finite element method, the sensitivity analysis performed usually by the help of the adjoint method, and an optimization method to minimize the objective function [43-45]. In this thesis, we will use the finite difference method which requires less implementation effort than the use of the finite element method.

The shape optimization procedure which will be considered in this thesis consists of three steps:

(1) To mesh the physical domain. The body–fitted (structured) grid generation technique will be used in this thesis to generate the grids over an arbitrarily shaped physical domain. The concepts of the structured grid generation methods and their formulations will be given in the next chapter.

(2) To solve the partial differential equations (PDE) describing the problem using the finite difference method as in [9, 46-53].

(3) To apply an appropriate optimization method to minimize the objective function.

Two optimization methods have been considered for the heat transfer part of this thesis: the Conjugate Gradient Method (CGM) and Levenberg-Marquardt (LM) one. As will be shown in next chapters, the conjugate gradient method is the preferable method to infer the optimal shape design.

1.8 Shape optimization in heat transfer: The State of the Art

In References [45, 54-56], two-dimensional optimal shape design in heat transfer under various boundary conditions has been successfully investigated. The utilized procedure in 2-D problems was mainly based on the application of three steps, namely boundary-fitted grid generation to mesh the physical domain and solve for the field variable (temperature), a sensitivity analysis scheme based on the perturbation method to compute the sensitivity of the temperatures on the boundary to variation of grid nodes positions, and conjugate gradient algorithm as an optimization method to minimize the difference between temperatures on the specified boundary and desired temperature. The sensitivity analysis scheme employed in the studies cited above is inferred by the perturbation method. This method can be error - prone and is computationally intensive. Furthermore, oscillations occur during the iteration process. A multistage operation, therefore, is required to infer optimal shape design. In other words, the solution process from the initial shape to the optimal one is divided into a few stages with appropriate intermediate conditions. In this procedure, the resulted optimal shape for each step is the initial shape for the next step, and the process is continued until the final optimal shape is obtained [56]. However, as will be shown in the next chapters, the sensitivity analysis approach adopted in this study is carried out using a novel method based on the finite difference method which results in an explicit expression for the calculation of the sensitivities. Furthermore, an effective and novel node redistribution method is proposed to detect, remove, and replace ill-ordered nodes to maintain the quality of mesh during the optimization process.

In Reference [57], Isogeometric analysis, a recently developed computational method based on integrating the finite element method into NURBS (non-uniform rational B-splines) proposed by Hughes et al. [58], is employed to represent the domain and solve the heat conduction equation. In

this approach, basis functions generated from NURBS are employed to construct an exact geometric model. The adjoint method is used to compute shape design sensitivity coefficients.

A geometry design of two-dimensional radiant enclosures using the Genetic Algorithm (a non-gradient algorithm) to create a desired heat flux distribution over the temperature-specified design surface is described in Reference [59]. The B-spline curves are used to provide a parametric representation of the boundary of the radiant enclosure. The geometry of the enclosure is designed by optimizing the position of control points representing the B-spline curves.

The level set method (one of the main approaches in structural shape and topology optimization) also is used in shape optimization problems in heat transfer [60-65]. In this method, domains and their boundaries are represented as level sets of a continuous function termed the level set function. A moving boundary is regarded as the zero level set. By the evolution of the shape, the corresponding level set function is determined by a Hamilton–Jacobi equation [66, 67].

Attention has also been given to three-dimensional shape optimization problems in heat transfer [68-70]. To the author's knowledge, no research is reported in 3-D shape optimization of heat conducting bodies using elliptic grid generation technique resulting in an explicit expression for the sensitivity analysis, and conjugate gradient with both Neumann and Robin boundary conditions. In References [68-70], the commercial codes CFD-RC and CFD-ACE+ have been employed to mesh the body and perform the sensitivity analysis. In addition, there are some limitations in these references. The first limitation is that the body shape is restricted to a cube. Another limitation of these references is the inability to handle Robin boundary conditions. Apart from the top face which is at constant temperature and the bottom face which is exposed to a heat flux, other faces are either insulated (Neumann boundary condition) or at constant temperature (Dirichlet boundary condition). The Robin boundary conditions are very common in heat transfer problems and the introduction of such boundary conditions allows a more comprehensive treatment on IHTPs. This study shows how to obtain the optimal shape design for both 3-D irregular and regular (cube) heat conducting bodies under different Neumann and Robin boundary conditions in an integrated and systematic manner.

In Reference [71], a three-dimensional optimal design problem using a general purpose commercial code (CFD-ACE+) and the Levenberg–Marquardt method is examined to estimate the optimal shape of an impingement cooling heat sink using different design variables. A three dimensional optimal shape design problem in simultaneously designing the shapes of interfacial surfaces among three conductive bodies, based on the desired system heat flux and domain volume, is examined in References [72, 73]. The Levenberg-Marquardt method, B-Spline surface generation technique together with the commercial code CFD-ACE+ were employed to solve the inverse geometry problem.

1.9 Motivations and novel contribution of this thesis

There are some key points which motivate a researcher to do research in the field of shape optimization. The objective of this study is to address and investigate them. The following highlights some of these points:

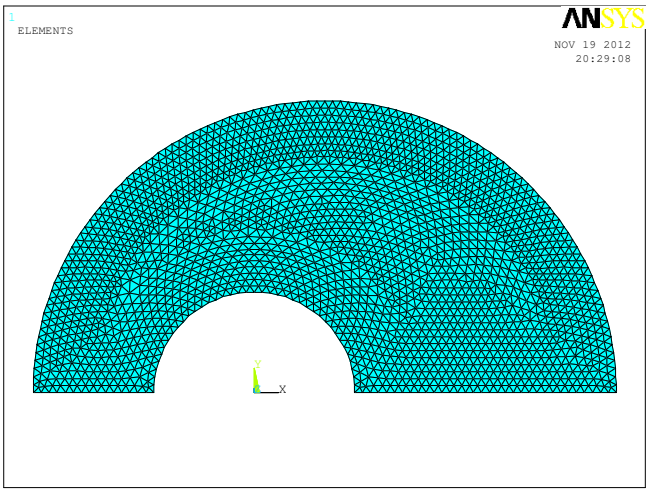
- ✓ Shape optimization is not restricted to the heat transfer problems. It has also gained much attention in fluid mechanics, the automobile industry, structural engineering, chemical engineering, and aerospace engineering to name a few. For instance, the reduction of the weight of a car while improving its performance, shape optimization of an aeroplane wing to reduce the effect of the drag acting on that, or the shape optimization of a structure to efficiently withstand given loads under certain constraints are a few samples taken from a significant number of shape optimization applications. This thesis aims to extend the procedure employed for the shape optimization in heat transfer problems to the aerodynamics shape optimization. Airfoil shape optimization to match the airfoil surface pressures to a desired distribution will be analysed in this thesis.
- ✓ Shape optimization is highly computationally intensive, highly time consuming and theoretically challenging. Therefore, this thesis also aims to propose algorithms which are more efficient and hence less computationally intensive and time consuming.

- ✓ Since the optimization procedure is an inseparable part of the shape optimization field and there exist various optimization methods (e.g., Gauss-Newton method, Conjugate Gradient method, Steepest Descent method, Levenberg-Marquardt method, and so forth), it is necessary to compare the relative merits of these methods.
- ✓ The sensitivity analysis also is a computationally intensive task and new methods which allow faster computations are required to accelerate the optimal shape design algorithms.
- ✓ The cornerstone of the numerical analysis is a grid generation scheme. This thesis also aims to study the effects of applying two different body–fitted grid generation methods (*Elliptic Grid Generation* and *Transfinite Interpolation Grid Generation*) on the efficiency of shape optimization.
- ✓ To study the feasibility of applying the derived shape optimization algorithm in heat transfer to other disciplines such as aerodynamic shape optimization.
- ✓ And, to examine the feasibility of applying the above points to 3D cases.

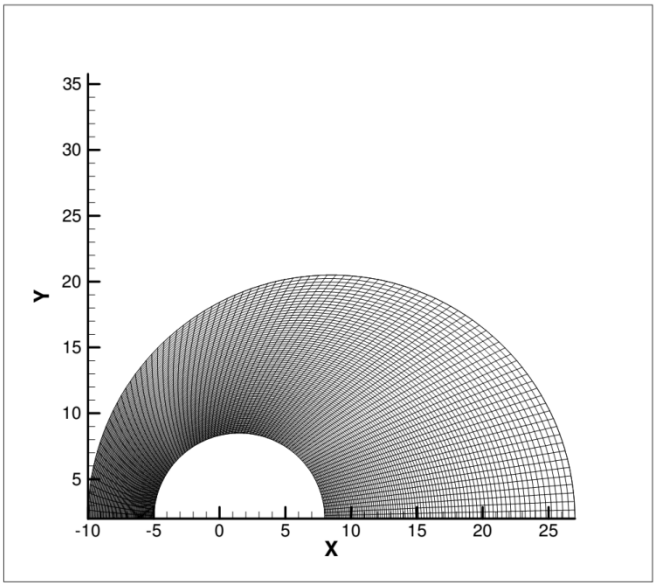
2. Two Dimensional Grid Generation and Direct Solver

2.1 Introduction

Heat transfer problems involving relatively simple geometries such as a sphere or a cylinder can be solved analytically. However, many practical heat transfer problems involve complicated geometries which cannot be solved analytically. In these situations, one seeks adequately accurate approximate solutions of the governing partial differential equations using numerical methods [52]. Grid generation is at the cornerstone of the numerical solution of partial differential equations. Indeed, in order to discretize the physical domain and approximating the PDEs by algebraic ones, a grid needs to be generated over the physical domain (body). Grids can be *structured* (quadrilateral cells in 2D or hexahedral cells in 3D) or *unstructured* (triangles in 2D and tetrahedral in 3D). The structured grids are also called curvilinear ones. Figure 2-1 shows structured and unstructured grids.



a) Unstructured grid generated in ANSYS



b) Structured grid generated using elliptic grid generation technique

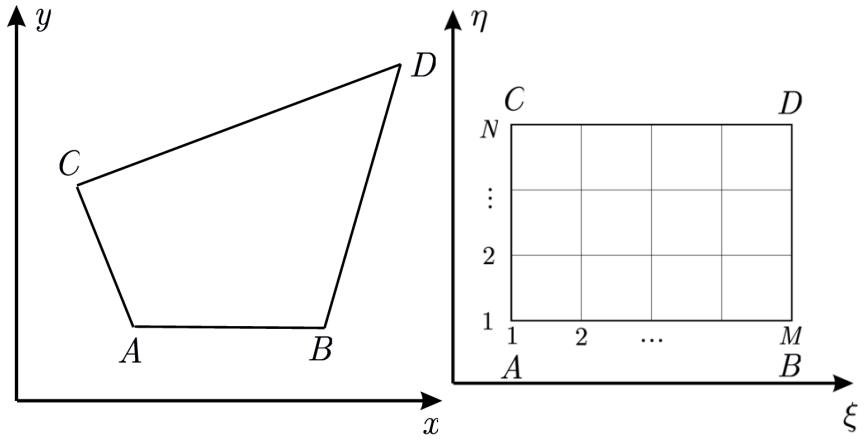
Figure 2-1 Unstructured and structured grids

Two structured grid generation methods considered in this study are *Elliptic Grid Generation* (PDE) and *Transfinite Interpolation Grid Generation* (Algebraic). Elliptic grid generation was proposed by Thompson [74] and is based on solving a system of elliptic partial differential equations to distribute nodes in the interior of the physical domain. This type of grid generator gives rise to grids which are uniformly distributed, even for non-smooth boundaries. Structured grid generation methods have been extensively studied in references [75-78]. Three unstructured grid generation procedures mainly used for the finite element method are the *Delaunay triangulation*, the *advancing front method*, and the *finite octree method* [79]. Unstructured grid generations techniques are covered in the course *computational geometry* [76, 78, 80-82] which is not within the scope of the thesis.

2.2 Elliptic Grid Generation

In order to obtain the numerical solution of a 2D heat conduction problem with a regular geometry, the finite difference method (FDM) has a major advantage: its computational simplicity enables us to generate uniformly

distributed grids over the regular physical domain. However, it is challenging to extend the method when the physical domain has a complicated geometry. The FDM is unable to handle effectively the numerical solution of the problem in this case. In such a situation, elliptic grid generation may be used to map the irregular physical domain from the x and y physical plane onto the ξ and η computational plane which is a regular region (see Figure 2-2).



a) Physical coordinate b) Computational coordinate

Figure 2-2 The irregular physical domain and the regular computational one. The physical domain (a) is mapped onto the computational one (b) to perform the calculation required to generate a grid over the physical domain.

On the other hand, Finite Element Method (FEM) can successfully handle complicated geometries. In FEM, a variety of appropriate mesh elements may be used depending on the problem under consideration, such as triangular and quadrilateral ones [46]. In contrast, mesh elements are in form of only quadrilateral for the FDM (whether using the traditional FDM or the FDM technique based on elliptic grid generation). However, the capability of FEM to treat complex geometries comes with the complexity of coding in comparison to FDM coding. In this thesis, an efficient and accurate FDM based method is proposed to infer the optimal shape for shape optimization problems in heat transfer and aerodynamics without the need for high computer storage and time spent developing and running the program. However, some key questions may arise: Is the use of boundary-fitted grid

generation with FDM accurate? How efficient is the use of boundary-fitted grids combined with FDM? What are advantages of this technique? These are the questions this chapter attempts to address.

Some preliminaries are required to derive the relations needed for generating grids over physical domains. Suppose we seek a relation (transformation) to transform a PDE from the independent variables x and y to the independent variables ξ and η (for a two dimensional case).This transformation can be stated as

$$\xi \equiv \xi(x, y) \tag{2.1}$$

$$\eta \equiv \eta(x, y) \tag{2.2}$$

and the inverse transformation is given as below

$$x \equiv x(\xi, \eta) \tag{2.3}$$

$$y \equiv y(\xi, \eta) \tag{2.4}$$

In engineering and science we often encounter PDEs in fields as varied as acoustics, heat transfer, fluid flow, electrostatics, electrodynamics, or elasticity. Since these PDEs involve first or second derivatives, relationships are needed to transform such derivatives from the (x, y) system to the $\left(\xi, \eta\right)$ one. In order to do this, the Jacobian of the transformation is needed which is given below

$$2\mathrm{D} : J = J\left(\frac{x, y}{\xi, \eta}\right) = \begin{vmatrix} x_\xi & y_\xi \\ x_\eta & y_\eta \end{vmatrix} = x_\xi y_\eta - x_\eta y_\xi \neq 0 \tag{2.5}$$

In three-dimensions, the Jacobian of transformation from the independent variables x, y, and z to the independent variables ξ, η, and ζ, respectively, is as following

$$3\mathrm{D} : J = J\left(\frac{x, y, z}{\xi, \eta, \zeta}\right) = \begin{vmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & z_\zeta \end{vmatrix}$$

$$= x_\xi \left(y_\eta z_\zeta - y_\zeta z_\eta\right) - y_\xi \left(x_\eta z_\zeta - x_\zeta z_\eta\right) + z_\xi (x_\eta y_\zeta - x_\zeta y_\eta) \neq 0 \tag{2.6}$$

As will be shown, the transformation relations involve the Jacobian in denominator. Hence, it cannot be zero. In this chapter, only the two-dimensional case is considered. Since in following sections we will deal with the Laplace equation in heat transfer, it is necessary to find relationships for the transformation of the first and second derivatives of the field variable T with respect to the position variables x and y . By using the chain rule, it can be concluded that

$$\frac{\partial T}{\partial x} = \frac{\partial T}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial T}{\partial \eta}\frac{\partial \eta}{\partial x} = \frac{\partial T}{\partial \xi}\xi_x + \frac{\partial T}{\partial \eta}\eta_x \tag{2.7}$$

$$\frac{\partial T}{\partial y} = \frac{\partial T}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial T}{\partial \eta}\frac{\partial \eta}{\partial y} = \frac{\partial T}{\partial \xi}\xi_y + \frac{\partial T}{\partial \eta}\eta_y \tag{2.8}$$

By interchanging x and ξ, and y and η, the following relationships can also be derived

$$\frac{\partial T}{\partial \xi} = \frac{\partial T}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial T}{\partial y}\frac{\partial y}{\partial \xi} = \frac{\partial T}{\partial x}x_\xi + \frac{\partial T}{\partial y}y_\xi \tag{2.9}$$

$$\frac{\partial T}{\partial \eta} = \frac{\partial T}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial T}{\partial y}\frac{\partial y}{\partial \eta} = \frac{\partial T}{\partial x}x_\eta + \frac{\partial T}{\partial y}y_\eta \tag{2.10}$$

By solving Equations (2.9) and (2.10) for $\frac{\partial T}{\partial x}$ and $\frac{\partial T}{\partial y}$, we finally obtain

$$\frac{\partial T}{\partial x} = \frac{1}{J}(y_\eta \frac{\partial T}{\partial \xi} - y_\xi \frac{\partial T}{\partial \eta}) \tag{2.11}$$

$$\frac{\partial T}{\partial y} = \frac{1}{J}(-x_\eta \frac{\partial T}{\partial \xi} + x_\xi \frac{\partial T}{\partial \eta}) \tag{2.12}$$

where $J = x_\xi y_\eta - x_\eta y_\xi$ is Jacobian of the transformation. By comparing Equations (2.7) and (2.11), and (2.8) and (2.12), it can be shown that

$$\xi_x = \frac{1}{J}y_\eta, \xi_y = -\frac{1}{J}x_\eta \tag{2.13}$$

$$\eta_x = -\frac{1}{J}y_\xi, \eta_y = \frac{1}{J}x_\xi \tag{2.14}$$

To transform terms in the Laplace equation (this is necessary for following chapters), the second order derivatives are needed. Therefore,

In the physical domain (x, y): $\nabla^2 T = \dfrac{\partial^2 T}{\partial x^2} + \dfrac{\partial^2 T}{\partial y^2} = 0$

After transformation, in the computational domain (ξ, η):

$$\nabla^2 T = \frac{1}{J^2}\Big(\alpha T_{\xi\xi} - 2\beta T_{\xi\eta} + \gamma T_{\eta\eta}\Big) + [(\nabla^2 \xi) T_{\xi} + \Big(\nabla^2 \eta\Big) T_{\eta}] \tag{2.15}$$

where

$$\alpha = x_{\eta}^{\ 2} + y_{\eta}^{\ 2}$$

$$\beta = x_{\xi} x_{\eta} + y_{\xi} y_{\eta}$$

$$\gamma = x_{\xi}^{\ 2} + y_{\xi}^{\ 2}$$

$$J = x_{\xi} y_{\eta} - x_{\eta} y_{\xi} (\text{Jacobian of transformation})$$

and $\nabla^2 \xi = P$ and $\nabla^2 \eta = Q$ are control functions which are calculated by introducing the intermediate variables R and S[47]

$$R = -\frac{1}{J^2}[\Big(x_{\eta}^2 + y_{\eta}^2\Big) x_{\xi\xi} + \Big(x_{\xi}^2 + y_{\xi}^2\Big) x_{\eta\eta} - 2\Big(x_{\xi} x_{\eta} + y_{\xi} y_{\eta}\Big) x_{\xi\eta}] \tag{2.16}$$

$$S = -\frac{1}{J^2}[\Big(x_{\eta}^2 + y_{\eta}^2\Big) y_{\xi\xi} + \Big(x_{\xi}^2 + y_{\xi}^2\Big) y_{\eta\eta} - 2\Big(x_{\xi} x_{\eta} + y_{\xi} y_{\eta}\Big) y_{\xi\eta}] \tag{2.17}$$

Finally, P and Q are given by

$$P = \frac{1}{J}(y_{\eta} R - x_{\eta} S) \tag{2.18}$$

$$Q = \frac{1}{J}(x_{\xi} S - y_{\xi} R) \tag{2.19}$$

All the necessary relationships to write the equations for the elliptic PDE grid generation have now been derived. Transformation of any PDE such as the continuity equation from the physical domain to the computational one can

be performed with the help of the above formulas. Here, only the relationships needed for elliptic grid generation involved in mapping from the physical domain to the computational one are considered. Thompson [74] proposed a grid generation technique based on solving the Poisson equations below

$$\xi_{xx} + \xi_{yy} = P\left(\xi,\eta\right) \tag{2.20}$$

$$\eta_{xx} + \eta_{yy} = Q(\xi,\eta) \tag{2.21}$$

where ξ and η are computational coordinates corresponding to x and y in the physical coordinate, respectively. P and Q are *grid control functions* which control the density of grids towards a specified coordinate line or about a specific grid point. To find an explicit relation for x and y in terms of grid points $\xi_i(i \in [1,M])$ and $\eta_j(j \in [1,N])$, the following relations may be used [75]

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2(P\left(\xi,\eta\right)x_{\xi} + Q\left(\xi,\eta\right)x_{\eta}) \tag{2.22}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2(P\left(\xi,\eta\right)y_{\xi} + Q\left(\xi,\eta\right)y_{\eta}) \tag{2.23}$$

where

$$\alpha = x_{\eta}^{\ 2} + y_{\eta}^{\ 2}$$

$$\beta = x_{\xi}x_{\eta} + y_{\xi}y_{\eta}$$

$$\gamma = x_{\xi}^{\ 2} + y_{\xi}^{\ 2}$$

$$J = x_{\xi}y_{\eta} - x_{\eta}y_{\xi}(\text{Jacobian of transformation}) \tag{2.24}$$

The solution of the above equations give x and y coordinates (in physical domain) of coordinate $\left(i,j\right)$ in the computational domain. In other words, $x\left(2,3\right) = k$ (k is a constant), means, for example, that the grid point $(2,3)$ in the computational coordinate has $x-$ coordinate of value k in the physical coordinate. The $x-$ and $y-$ coordinates of the grid points at the corners of the computational domain remain constant and are equal to corresponding ones in physical coordinate, as illustrated in Figure 2-2. Therefore,

$$x_{1,1}|_{\text{comp. domain}} = x_A|_{\text{phys. domain}} \quad , \quad x_{M,1}|_{\text{comp. domain}} = x_B|_{\text{phys. domain}}$$

$$x_{1,N}|_{\text{comp. domain}} = x_C|_{\text{phys. domain}} \quad , \quad x_{M,N}|_{\text{comp. domain}} = x_D|_{\text{phys. domain}}$$

$$y_{1,1}|_{\text{comp. domain}} = y_A|_{\text{phys. domain}} \quad , \quad y_{M,1}|_{\text{comp. domain}} = y_B|_{\text{phys. domain}}$$

$$y_{1,N}|_{\text{comp. domain}} = y_C|_{\text{phys. domain}} \quad , \quad y_{M,N}|_{\text{comp. domain}} = y_D|_{\text{phys. domain}}$$

Using the finite difference method, Equations (2.22) through (2.24) can be discretized. Assuming $\Delta\eta = \Delta\xi = 1$,

$$(x_\xi)_{i,j} = \frac{1}{2}(x_{i+1,j} - x_{i-1,j}), (x_\eta)_{i,j} = \frac{1}{2}(x_{i,j+1} - x_{i,j-1})$$

$$(y_\xi)_{i,j} = \frac{1}{2}(y_{i+1,j} - y_{i-1,j}), (y_\eta)_{i,j} = \frac{1}{2}(y_{i,j+1} - y_{i,j-1})$$

$$\left(x_{\xi\xi}\right)_{i,j} = (x_{i+1,j} - 2x_{i,j} + x_{i-1,j}), \left(y_{\xi\xi}\right)_{i,j} = (y_{i+1,j} - 2y_{i,j} + y_{i-1,j})$$

$$\left(x_{\eta\eta}\right)_{i,j} = (x_{i,j+1} - 2x_{i,j} + x_{i,j-1}), \left(y_{\eta\eta}\right)_{i,j} = (y_{i,j+1} - 2y_{i,j} + y_{i,j-1})$$

$$(x_{\xi\eta})_{i,j} = \frac{1}{4}(x_{i+1,j+1} - x_{i-1,j+1} - x_{i+1,j-1} + x_{i-1,j-1})$$

$$(y_{\xi\eta})_{i,j} = \frac{1}{4}(y_{i+1,j+1} - y_{i-1,j+1} - y_{i+1,j-1} + y_{i-1,j-1}) \tag{2.25}$$

The above relationships are valid for all interior nodes, i.e., $i \in [2, M-1]$ and $j \in [2, N-1]$, where M and N are the number of nodes in the ξ and η directions, respectively. For boundary grid points along $\xi = 1$, $\xi = M$, $\eta = 1$, and $\eta = N$, we can simply impose a uniform distribution of the nodes. In other words, $(M-1)$ equal intervals on sides AB and CD and $(N-1)$ equal intervals on sides AC and BD.

A non-linear set of algebraic equations are then obtained in terms of $x_{i,j}$ and $y_{i,j}$ by substituting the relations (2.25) into differential equations (2.22) and (2.23) and assuming $P = Q = 0$ for generating a smooth grid over the physical domain. In order to solve these equations, an iterative method

such as *Successive Over Relaxation* (SOR) may be used due to its high convergence rate relative to other iterative methods such as the Gauss-Seidel method. If we retain $x_{i,j}$ and $y_{i,j}$ with coefficient $+1$ in left side of equation and take other terms involved in Equations (2.22) and (2.23) to the right side (that is, $x_{i,j} = f(x)$, and $y_{i,j} = g(y)$), by using SOR in form of

$$x_{i,j}^{(\mathrm{k})} = \omega x_{i,j}^{(\mathrm{k}-1)} + (1-\omega) x_{i,j}^{(\mathrm{k}-2)} \tag{2.26}$$

$$y_{i,j}^{(\mathrm{k})} = \omega y_{i,j}^{(\mathrm{k}-1)} + (1-\omega) y_{i,j}^{(\mathrm{k}-2)} \tag{2.27}$$

x and y coordinates of grid points (i, j) can be obtained. In these relations for SOR, k is iteration number, and ω is *relaxation factor* which has a value of $1 < \omega < 2$. A value of $\omega = 1.8$ is chosen in this thesis.

By programming all the finite difference relationships for a given physical domain in MATLAB or, if a very fine grid is needed, in Fortran or C++, one can generate boundary-fitted (elliptic) grids. Other PDE grid generators (namely, parabolic and hyperbolic) are studied in reference [76]. The difference between the PDE grid generators stems from the form of the PDE to be solved. These grid generators have advantages and disadvantages including:

Hyperbolic grid generator:

- *Advantage*: much faster than the elliptic one; it prevents the grid line overlapping at concave boundaries.
- *Disadvantage*: propagation of the boundary discontinuities into the domain.

Parabolic grid generator: Its features are a compromise between the hyperbolic and elliptic ones.

- *Advantage*: no propagation of the boundary discontinuities into the domain, as in elliptic one; it is fast, as in hyperbolic one.
- *Disadvantage*: Implementation of the equations is not straightforward; the expressions for the control function are not developed [47].

Some examples of elliptic grid generation for different geometries and different grid sizes are plotted in Figure 2-3.

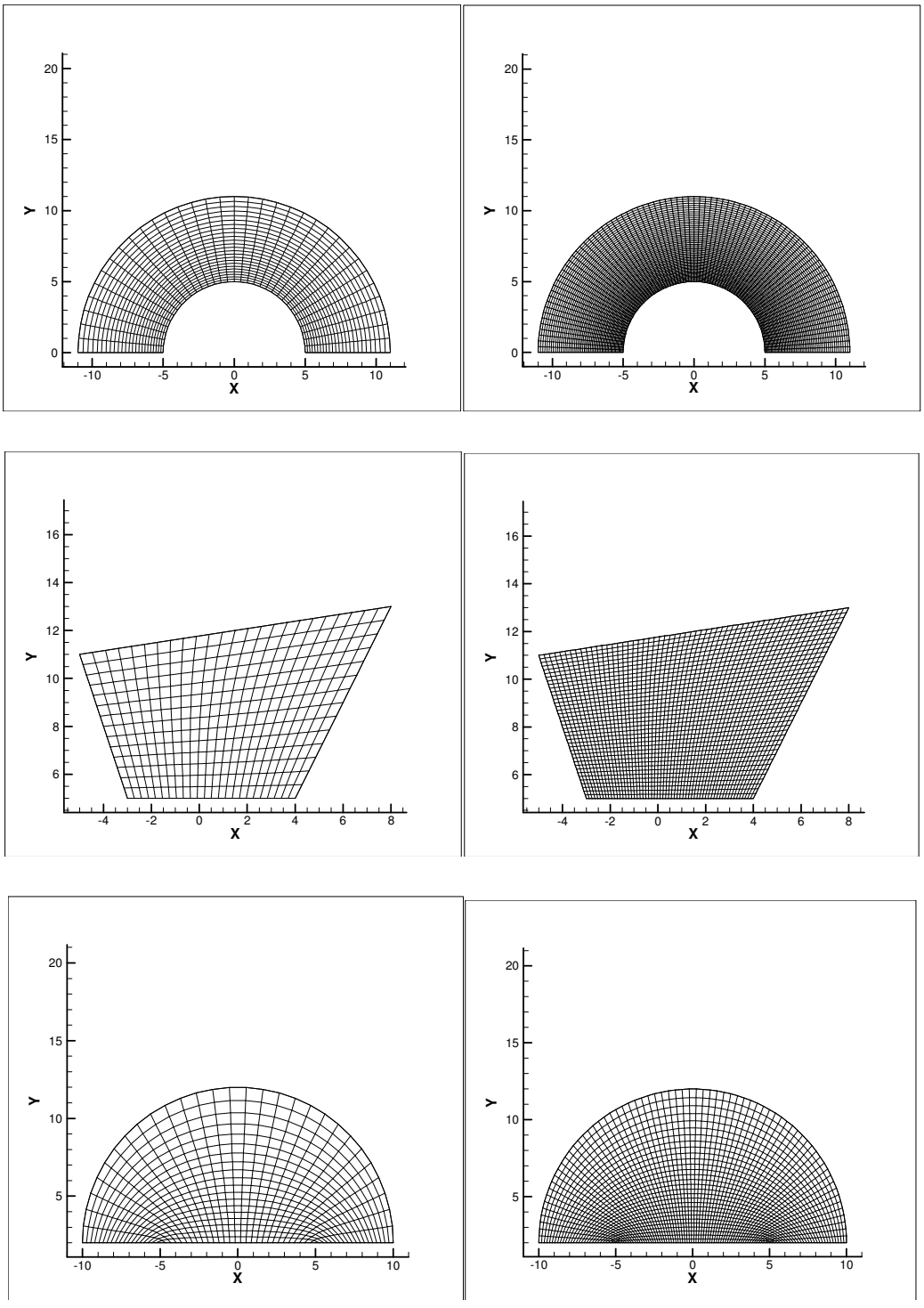
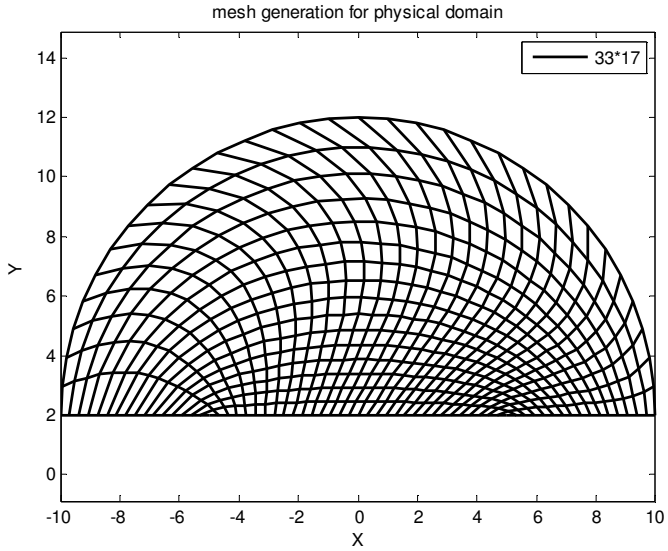


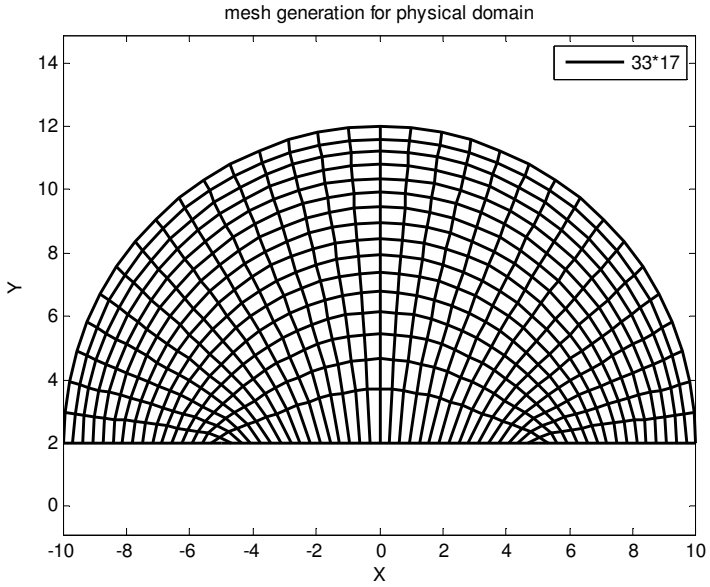
Figure 2-3 The grids with different geometries and sizes generated by the elliptic grid generation technique.

To illustrate the effect of P and Q, the grid control functions, on the resulting grid, tests with different values for P and Q are explored

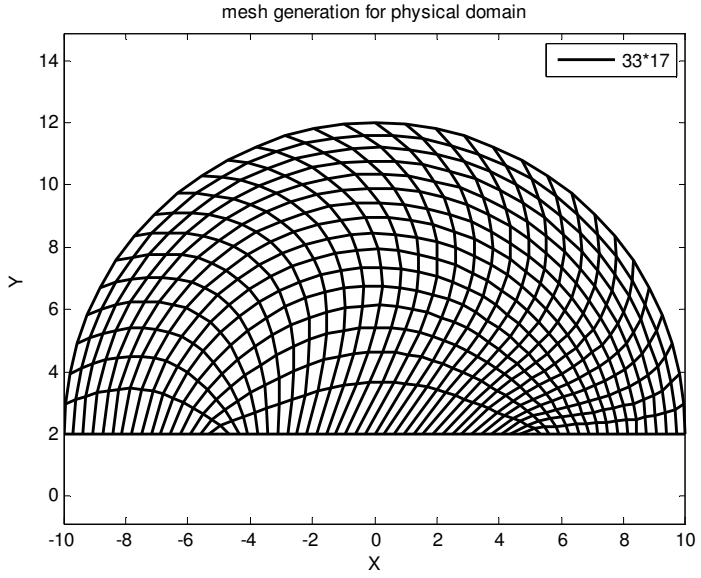
$$P = 0.45, Q = 0.0$$



$$P = 0.0, Q = 0.45$$



$$P = 0.45, Q = 0.45$$



$$P = 0.0, Q = -0.25$$

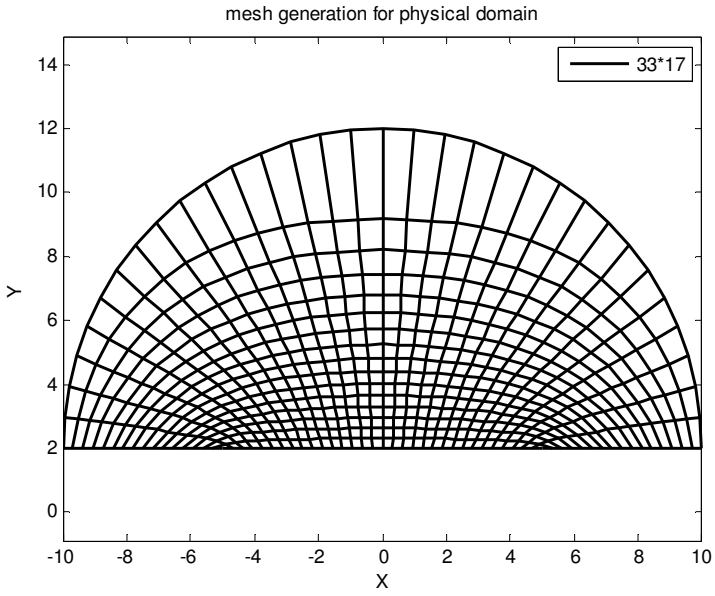


Figure 2-4 effect of grid control functions on the resulting grids.

In other words, as shown in Figure 2-1b, depending on whether the values of P and Q are positive or negative, the lines of constant ξ and η are attracted towards the right- or left-hand side, and up or down, respectively. Expressions for P and Q to suit particular meshing requirement are reported in references [46, 74-76, 78]. As pointed out previously, the special case $P = Q = 0$ leads to a smooth grid which is shown in the following figure.

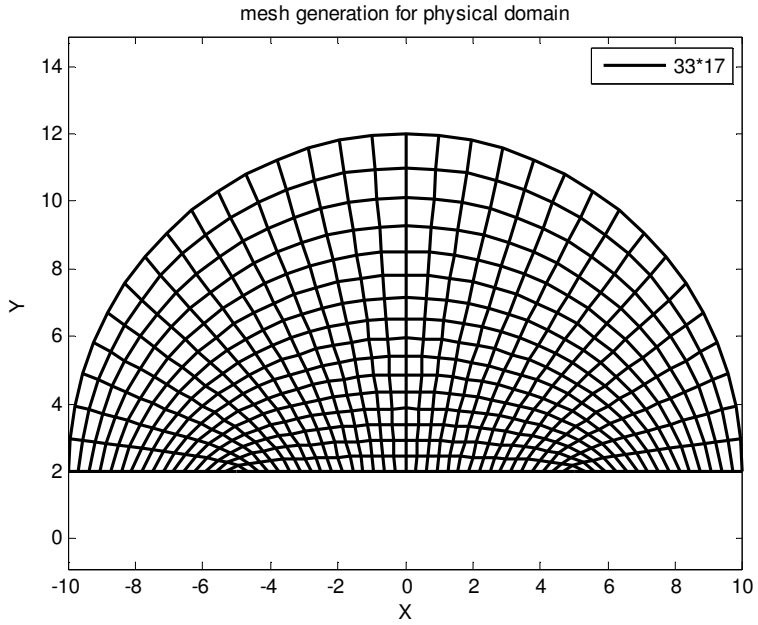


Figure 2-5 a grid with control functions $P = Q = 0$.

2.3 Algebraic Grid Generation (TransFinite Interpolation -TFI)

TFI is another powerful structured grid generation which is based on interpolating from the boundaries of the physical domain. This method initially proposed by Gordon and Hall [83] uses unidirectional interpolation functions such as Lagrange or Hermite polynomials, or Spline functions. In two directions, a linear Lagrange interpolation functions for each of ξ and η coordinates can be written separately. If the linear Lagrange polynomials $1-\xi$, ξ, $1-\eta$ and η are used as blending functions for the interval $[0, 1]$ for the direction of ξ and the interval $[0, 1]$ for the direction of η, the TFI

formula for the interval $\left[1,M\right]$ for the direction of ξ and the interval $\left[1,N\right]$ for the direction of η is given below [47, 75-78, 83, 84] (see Figure 2-6)

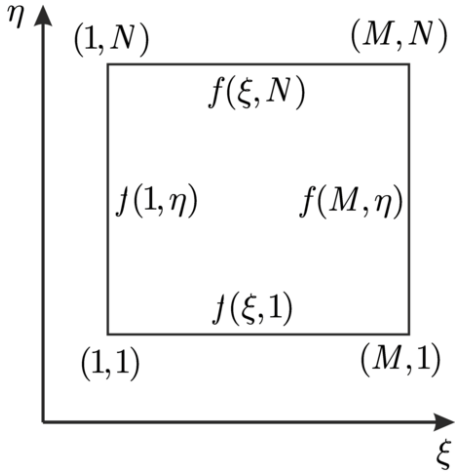


Figure 2-6 Computational domain used in TFI with $f(\xi,\eta)$ specified on the lines of constant ξ and η.

$$f\left(\xi,\eta\right)=\left(x\left(\xi,\eta\right),y(\xi,\eta)\right)$$

$$f\left(\xi,\eta\right)=f(\xi,\eta)^{(1)}+B_{1(\eta)}\left[f\left(\xi,1\right)-f\left(\xi,1\right)^{(1)}\right]+B_{2(\eta)}[f\left(\xi,N\right)-f\left(\xi,N\right)^{(1)}] \quad (2.28)$$

where

$$f(\xi,\eta)^{(1)}=A_{1(\xi)}f\left(1,\eta\right)+A_{2(\xi)}f(M,\eta) \quad (2.29)$$

$$A_{1(\xi)}=\frac{M-\xi}{M-1},A_{2(\xi)}=1-A_{1(\xi)}=1-\frac{M-\xi}{M-1}=\frac{M-1-M+\xi}{M-1}=\frac{\xi-1}{M-1}$$

$$B_{1(\eta)}=\frac{N-\eta}{N-1},B_{2(\eta)}=\frac{\eta-1}{N-1} \quad (2.30)$$

Since $f=x$ and y, for x we have

$$x(\xi,\eta)^{(1)}=A_{1(\xi)}x\left(1,\eta\right)+A_{2(\xi)}x\left(M,\eta\right)$$

$$=\frac{M-\xi}{M-1}x\left(1,\eta\right)+\frac{\xi-1}{M-1}x(M,\eta) \quad (2.31)$$

$$x\big(\xi,\eta\big)=x\big(\xi,\eta\big)^{(1)}+\frac{N-\eta}{N-1}\Big[x\big(\xi,1\big)-x\big(\xi,1\big)^{(1)}\Big]$$

$$+\frac{\eta-1}{N-1}\Big[x\big(\xi,N\big)-x\big(\xi,N\big)^{(1)}\Big] \tag{2.32}$$

Hence,

$$x\big(\xi,\eta\big)=\frac{M-\xi}{M-1}x\big(1,\eta\big)+\frac{\xi-1}{M-1}x\big(M,\eta\big)+\frac{N-\eta}{N-1}\Big[x\big(\xi,1\big)-x\big(\xi,1\big)^{(1)}\Big]$$

$$+\frac{\eta-1}{N-1}\Big[x\big(\xi,N\big)-x\big(\xi,N\big)^{(1)}\Big] \tag{2.33}$$

And we get

$$x\big(\xi,\eta\big)=\frac{M-\xi}{M-1}x\big(1,\eta\big)+\frac{\xi-1}{M-1}x\big(M,\eta\big)$$

$$+\frac{N-\eta}{N-1}\Big[x\big(\xi,1\big)-(\frac{M-\xi}{M-1}x\big(1,1\big)+\frac{\xi-1}{M-1}x(M,1))\Big]$$

$$+\frac{\eta-1}{N-1}\Big[x\big(\xi,N\big)-(\frac{M-\xi}{M-1}x\big(1,\eta\big)+\frac{\xi-1}{M-1}x(M,\eta))\Big] \tag{2.34}$$

After simplifying

$$x\big(\xi,\eta\big)=\frac{M-\xi}{M-1}x\big(1,\eta\big)+\frac{\xi-1}{M-1}x\big(M,\eta\big)$$

$$+\frac{N-\eta}{N-1}\Big[x\big(\xi,1\big)-\frac{M-\xi}{M-1}x\big(1,1\big)-\frac{\xi-1}{M-1}x(M,1)\Big]$$

$$+\frac{\eta-1}{N-1}\Big[x\big(\xi,N\big)-\frac{M-\xi}{M-1}x\big(1,\eta\big)-\frac{\xi-1}{M-1}x(M,\eta)\Big] \tag{2.35}$$

By changing x to y,

$$y\big(\xi,\eta\big)=\frac{M-\xi}{M-1}y\big(1,\eta\big)+\frac{\xi-1}{M-1}y\big(M,\eta\big)$$

$$+\frac{N-\eta}{N-1}\Big[y\big(\xi,1\big)-\frac{M-\xi}{M-1}y\big(1,1\big)-\frac{\xi-1}{M-1}y(M,1)\Big]$$

$$+\frac{\eta-1}{N-1}\left[y\left(\xi,N\right)-\frac{M-\xi}{M-1}y\left(1,\eta\right)-\frac{\xi-1}{M-1}y(M,\eta)\right] \tag{2.36}$$

By coding these two relationships for interior nodes (boundaries all are known and mapped without change) as following

$$x_{i,j}=\frac{M-i}{M-1}x_{1,j}+\frac{i-1}{M-1}x_{M,j}+\frac{N-j}{N-1}\left[x_{i,1}-\frac{M-i}{M-1}x_{1,1}-\frac{i-1}{M-1}x_{M,1}\right]$$

$$+\frac{j-1}{N-1}\left[x_{i,N}-\frac{M-i}{M-1}x_{1,j}-\frac{i-1}{M-1}x_{M,j}\right] \tag{2.37}$$

$$y_{i,j}=\frac{M-i}{M-1}y_{1,j}+\frac{i-1}{M-1}y_{M,j}+\frac{N-j}{N-1}\left[y_{i,1}-\frac{M-i}{M-1}y_{1,1}-\frac{i-1}{M-1}y_{M,1}\right]$$

$$+\frac{j-1}{N-1}\left[y_{i,N}-\frac{M-i}{M-1}y_{1,j}-\frac{i-1}{M-1}y_{M,j}\right] \tag{2.38}$$

$$i\in\left[2,M-1\right],j\in[2,N-1]$$

a boundary-fitted grid can be generated over the physical domain. Some numerical results for TFI grid generation are shown below.

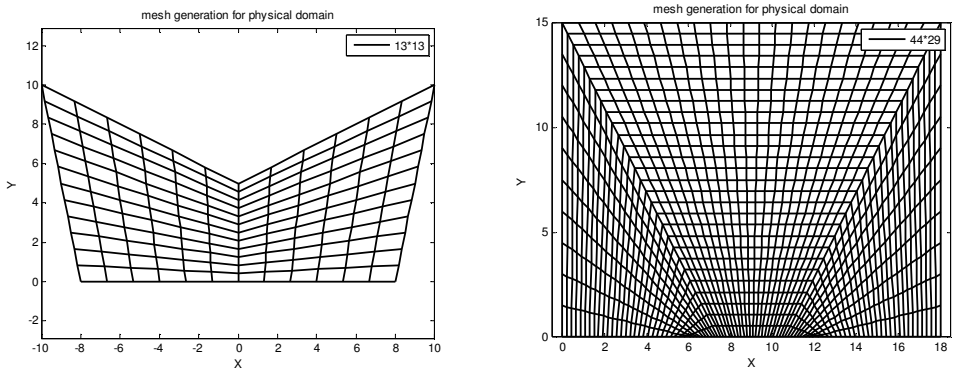


Figure 2-7 grids based on TFI grid generation

The TFI has disadvantages such as a lack of smoothness and folding grids in complicated physical domains. If there is a slope discontinuity on the any boundary of the physical domain, it will propagate into the interior of the

domain [77, 78]. Contrary to TFI, elliptic grid generation generates smooth grids over the physical domain. TFI is an algebraic method and hence it does not require an iterative procedure to obtain the results in contrast to the elliptic grid generation. Therefore, from computational viewpoint, TFI gives results faster than elliptic grid generation, especially for very fine grids. The ability of these two structured grid generation methods in solving heat transfer problems and shape optimization will be compared in the following. For the same domains and grid size as domains in Figure 2-7, elliptic grid generation gives the following results:

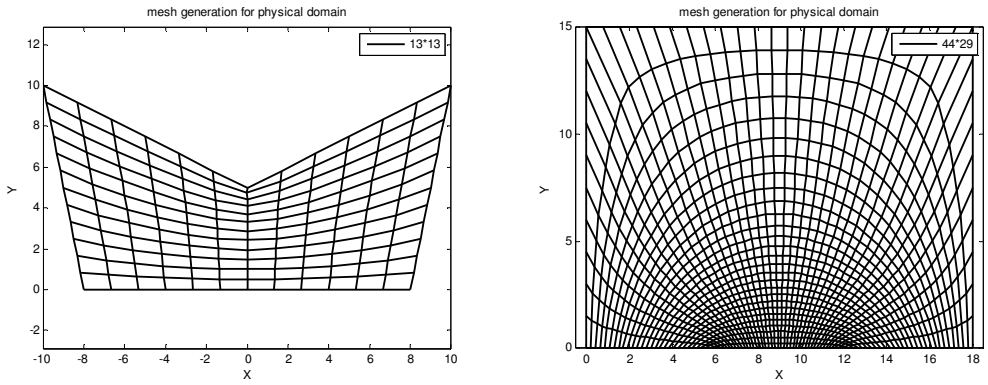


Figure 2-8 grids based on elliptic grid generation

2.4 Direct solver for the field equation

The elliptic grid generation technique and the steady heat conduction equation both satisfy Laplace equation and therefore much of the effort put into programming the relations required to generate the grid over the physical domain can be reused to solve the heat transfer problem. Consider the 2D steady state heat conduction equation

$$T_{xx} + T_{yy} = 0 \tag{2.39}$$

in the physical domain shown in Figure 2-9 with the following combination of the *Neumann* and *Robin* boundary conditions.

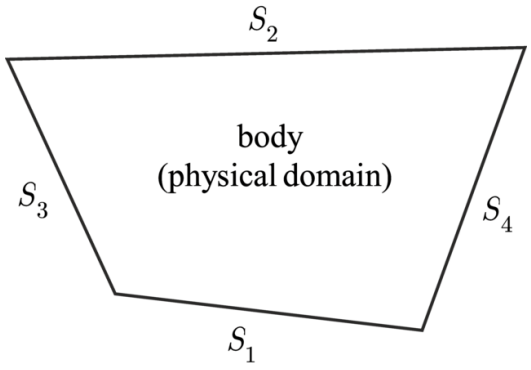


Figure 2-9 an irregular four-sided physical domain

On surface S_1 a heat flux $= \dot{q}_1$ is imposed, convective heat transfer is imposed on surfaces $S_i (i = 2,3,4)$ with corresponding heat transfer coefficients $h_i (i = 2,3,4)$ and surrounding temperatures $T_{\infty_i} (i = 2,3,4)$, respectively. Now the heat conduction equation and corresponding boundary conditions equations should be transformed from the x, y physical coordinates to the ξ, η computational ones. The transformation results in

$$\alpha T_{\xi\xi} - 2\beta T_{\xi\eta} + \gamma T_{\eta\eta} = -J^2 (P(\xi,\eta) T_\xi + Q(\xi,\eta) T_\eta) \tag{2.40}$$

where

$$\alpha = x_\eta^{\,2} + y_\eta^{\,2}$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta$$

$$\gamma = x_\xi^{\,2} + y_\xi^{\,2}$$

$$J = x_\xi y_\eta - x_\eta y_\xi \left(\text{Jacobian of transformation}\right) \tag{2.41}$$

If the boundary conditions are of the Dirichlet kind (such as temperature distribution), no transformation is needed at the boundary as the temperature values remain constant in the computational domain. However, if there are Neumann boundary conditions (such as heat flux on a boundary) or Robin boundary conditions (such as convective heat transfer on a boundary), a transformation is needed as the derivative of temperature (heat flux) in the physical domain is not the same as that in the computational domain. In fact,

$\frac{\partial T}{\partial n}$ at a boundary in the physical domain is related to $\frac{\partial T}{\partial \xi}$ and/or $\frac{\partial T}{\partial \eta}$ in the computational domain. To obtain such mathematical relationships, say for the surface S_4, the following formulations may be used:

$$\frac{\partial T}{\partial n} = \nabla T.\mathbf{n} \tag{2.42}$$

where $\mathbf{n}$ is the outward-pointing unit normal vector to a given surface.

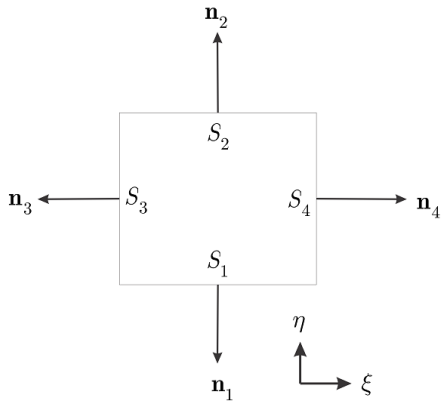


Figure 2-10 The outward – pointing unit normal vectors to $\xi =$ constant and $\eta =$ constant lines.

As shown in Figure 2-10, for the surface S_4, $\mathbf{n} = \xi$. Therefore,

$$\frac{\partial T}{\partial \xi} = \nabla T \cdot \frac{\nabla \xi}{\left|\nabla \xi\right|} = (T_x \mathbf{i} + T_y \mathbf{j}) \cdot \left(\frac{\xi_x \mathbf{i} + \xi_y \mathbf{j}}{\sqrt{\xi_x^2 + \xi_x^2}}\right) \tag{2.43}$$

Substituting ξ_x and ξ_y from Equation (2.13), T_x and T_y from Equations (2.11) and (2.12), and taking the dot product of two vectors in Equation (2.43), we get

$$\frac{\partial T}{\partial \xi} = \left[\frac{1}{J}(y_\eta T_\xi - y_\xi T_\eta)\right]\frac{\frac{y_\eta}{J}}{\sqrt{(\frac{y_\eta}{J})^2 + (\frac{-x_\eta}{J})^2}} + \left[\frac{1}{J}(-x_\eta T_\xi + x_\xi T_\eta)\right]\frac{-\frac{x_\eta}{J}}{\sqrt{(\frac{y_\eta}{J})^2 + (\frac{-x_\eta}{J})^2}}$$

(2.44)

From Equation (2.41) we know that $\alpha = {x_\eta}^2 + {y_\eta}^2$. Therefore,

$$\frac{\partial T}{\partial \xi} = \left[\frac{1}{J}(y_\eta T_\xi - y_\xi T_\eta)\right]\frac{\frac{y_\eta}{J}}{\sqrt{\frac{\alpha}{J^2}}} + \left[\frac{1}{J}(-x_\eta T_\xi + x_\xi T_\eta)\right]\frac{-\frac{x_\eta}{J}}{\sqrt{\frac{\alpha}{J^2}}}$$

After simplifying,

$$\frac{\partial T}{\partial \xi} = \left[\frac{1}{J}(y_\eta T_\xi - y_\xi T_\eta)\right]\frac{y_\eta}{\sqrt{\alpha}} + \left[\frac{1}{J}(-x_\eta T_\xi + x_\xi T_\eta)\right]\frac{-x_\eta}{\sqrt{\alpha}}$$

$$= \frac{1}{J\sqrt{\alpha}}\Big[y_\eta^2 T_\xi - y_\eta y_\xi T_\eta + x_\eta^2 T_\xi - x_\eta x_\xi T_\eta\Big]$$

$$= \frac{1}{J\sqrt{\alpha}}\Big[T_\xi(y_\eta^2 + x_\eta^2) - T_\eta(x_\eta x_\xi + y_\eta y_\xi)\Big]$$

$$\frac{\partial T}{\partial \xi} = \frac{1}{J\sqrt{\alpha}}\Big[\alpha T_\xi - \beta T_\eta\Big] \tag{2.45}$$

By the similar approach, the equations for other surfaces can be obtained which are summarized as follows

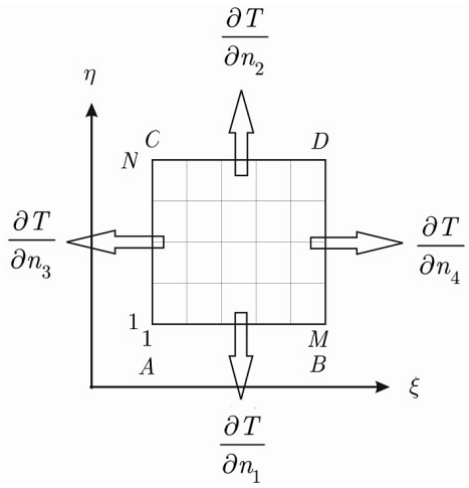


Figure 2-11 Neumann and Robin conditions at the boundaries of the computational domain

$$\text{on surface } S_1 : \frac{\partial T}{\partial n_1} = \frac{-1}{J\sqrt{\gamma}}(\gamma T_\eta - \beta T_\xi) \tag{2.46}$$

$$\text{on surface } S_2 : \frac{\partial T}{\partial n_2} = \frac{1}{J\sqrt{\gamma}}(\gamma T_\eta - \beta T_\xi) \tag{2.47}$$

$$\text{on surface } S_3 : \frac{\partial T}{\partial n_3} = \frac{-1}{J\sqrt{\alpha}}(\alpha T_\xi - \beta T_\eta) \tag{2.48}$$

$$\text{on surface } S_4 : \frac{\partial T}{\partial n_4} = \frac{1}{J\sqrt{\alpha}}(\alpha T_\xi - \beta T_\eta) \tag{2.49}$$

where the coefficients α, β, γ, and J are as defined by Equations (2.41). Using FDM, one can discretize the relations for T_η and T_ξ at every boundary with Neumann and Robin conditions [46]. For example, for nodes on surface S_2 with $i \in [2, M-1]$, it can be written

$$\frac{\partial T}{\partial \xi} = T_\xi = \frac{T_{i+1,N} - T_{i-1,N}}{2} \tag{2.50}$$

$$\frac{\partial T}{\partial \eta} = T_\eta = \frac{3T_{i,N} - 4T_{i,N-1} + T_{i,N-2}}{2} \tag{2.51}$$

By equating the heat conduction and convection equations on the surface S_2, we have

$$\dot{q}_{\text{conduction}} = \dot{q}_{\text{convection}}$$

$$-k\frac{\partial T}{\partial n_2} = h_2\left(T_{S_2} - T_{\infty_2}\right)$$

$$-k\frac{\partial T}{\partial \eta} = h_2\left(T_{S_2} - T_{\infty_2}\right)$$

$$-k\left[\frac{1}{J\sqrt{\gamma}}\left(\gamma T_{\eta} - \beta T_{\xi}\right)\right] = h_2\left(T_{i,N} - T_{\infty_2}\right) \tag{2.52}$$

By substituting the relations (2.50) and (2.51) into Equation (2.52) and then solving for $T_{i,N}$ using an algebraic equation solver such as Maple, a relationship is derived which allow us to obtain the temperature distribution on the surface S_2. The expression for $T_{i,N}$, therefore, is

$$T_{i,N} = \frac{1}{(h_2 + \frac{3k\sqrt{\gamma}}{2J})}\left[\frac{-k\sqrt{\gamma}}{2J}(-4T_{i,N-1} + T_{i,N-2}) + \frac{k\beta}{2J\sqrt{\gamma}}(T_{i+1,N} - T_{i-1,N}) + h_2 T_{\infty_2}\right] \tag{2.53}$$

Similar relationships may be found for other boundaries with Neumann conditions (prescribed heat flux) and Robin conditions (convective heat transfer). Furthermore, if there is radiation heat transfer on a given surface, say surface S_2, we can simply write the radiation equation on the right side of Equation (2.52) and then solve it for $T_{i,N}$ using the Newton-Raphson method. By considering Figure 2-9, on surface S_1 a heat flux $= \dot{q}_1$ is imposed, radiative heat transfer is imposed on surfaces $S_i (i = 2,3,4)$ with corresponding emissivities $\varepsilon_i (i = 2,3,4)$ and surrounding temperatures $T_{S_i} (i = 2,3,4)$, respectively. The procedure for implementing the radiation boundary conditions for nodes on surface S_2 with $i \in [2, M-1]$ can be expressed as follows

$$\dot{q}_{\text{conduction}} = \dot{q}_{\text{radiation}}$$

$$-k\frac{\partial T}{\partial n_2} = \varepsilon_2\sigma(T_{i,N}^4 - T_{S_2}^4)$$

$$-k\frac{\partial T}{\partial \eta} = \varepsilon_2\sigma(T_{i,N}^4 - T_{S_2}^4)$$

Using Equation (2.47) it can be written

$$-k\left[\frac{1}{J\sqrt{\gamma}}(\gamma T_\eta - \beta T_\xi)\right] = \varepsilon_2\sigma\left[(T_{i,N} + 273.15)^4 - (T_{S_2} + 273.15)^4\right] \qquad (2.54)$$

By substituting Equations (2.50) and (2.51) for surface S_2, we get

$$-k\left[\frac{1}{J\sqrt{\gamma}}(\gamma(\frac{1}{2})(3T_{i,N} - 4T_{i,N-1} + T_{i,N-2}) - \beta(\frac{1}{2})(T_{i+1,N} - T_{i-1,N}))\right]$$

$$= \varepsilon_2\sigma\left[(T_{i,N} + 273.15)^4 - (T_{S_2} + 273.15)^4\right] \qquad (2.55)$$

Equation (2.55) is a nonlinear equation (due to the fourth power at the right side) and should be solved for $T_{i,N}$. Therefore, Newton-Raphson method may be used as follows

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad (2.56)$$

It can be written for $T_{i,N}$ as below

$$T_{i,N}\big|_{n+1} = T_{i,N}\big|_n - \frac{f(T_{i,N})_n}{f'(T_{i,N})_n} \qquad (2.57)$$

where

$$f(T_{i,N}) = -k\left[\frac{1}{J\sqrt{\gamma}}(\gamma(\frac{1}{2})(3T_{i,N} - 4T_{i,N-1} + T_{i,N-2}) -$$

$$\beta(\frac{1}{2})(T_{i+1,N} - T_{i-1,N}))\right] - \varepsilon_2\sigma\left[(T_{i,N} + 273.15)^4 - (T_{S_2} + 273.15)^4\right] \qquad (2.58)$$

and

$$f'(T_{i,N}) = \frac{\partial f(T_{i,N})}{\partial T_{i,N}} = \frac{-3k\sqrt{\gamma}}{2J} - 4\varepsilon_2\sigma(T_{i,N} + 273.15)^3 \qquad (2.59)$$

Substituting Equations (2.58) and (2.59) into Equation (2.57) and solving for $T_{i,N}$, one can obtain the temperature of nodes on surface S_2, $T_{i,N}$, with $i \in [2, M-1]$. Similar approaches can result in computing the temperature of nodes on surfaces $S_i (i = 1,3,4)$.

2.5 Errors

It is clear that every computer has a limited number of digits during arithmetic calculations. With the use of *Double Precision* digits, one can increase the accuracy of numerical calculations. The difference between exact value of a number and its approximate one is called *Round-Off Error*. This error is directly proportional to the number of computations in a solution. By increasing the grid size, the number of computations also increases which results in increasing the round-off error. On the other hand, increasing the grid size leads to decrease the truncation error (*discretization error).* The magnitude of the total error is the sum of the round-off and truncation errors. Hence, the optimal values of M and N describing the grid size might be obtained to have the least total error. Furthermore, a suitable tolerance is required for iterative procedure to have appropriate computing time proportional to capabilities of machine.

2.6 Orthogonality of grid lines

In the field of computational simulation, the accuracy of numerical solutions is affected by the smoothness and orthogonality of grid lines. Deviation of grid spacing from orthogonality, particularly at the physical domain boundaries, can results in truncation error [85]. Therefore, orthogonality of the grid lines at the physical domain boundaries is required to obtain accurate results. In order to develop a mathematical expression to implement such a requirement, consider the gradient of ξ and η defined by

$$\nabla\xi = \frac{\partial \xi}{\partial x}\mathbf{i} + \frac{\partial \xi}{\partial y}\mathbf{j} \tag{2.60}$$

$$\nabla\eta = \frac{\partial \eta}{\partial x}\mathbf{i} + \frac{\partial \eta}{\partial y}\mathbf{j} \tag{2.61}$$

where $\mathbf{i}$ and $\mathbf{j}$ are unit vectors. The dot product of $\nabla\xi$ and $\nabla\eta$ is

$$\nabla\xi \cdot \nabla\eta = \xi_x \eta_x + \xi_y \eta_y \tag{2.62}$$

By substituting the metrics from Equations (2.13) and (2.14) into Equation (2.62), we obtain

$$\nabla\xi \cdot \nabla\eta = (\frac{1}{J} y_\eta)(-\frac{1}{J} y_\xi) + (-\frac{1}{J} x_\eta)(\frac{1}{J} x_\xi) = -\frac{1}{J^2}(x_\xi x_\eta + y_\xi y_\eta) \tag{2.63}$$

Knowing that the dot product of $\nabla\xi$ and $\nabla\eta$ is equal to the cosine of the angle ϕ(see Figure 2-12), we have

$$\cos\phi = -\frac{1}{J^2}(x_\xi x_\eta + y_\xi y_\eta) \tag{2.64}$$

To have orthogonality condition at the boundaries, the angle ϕ should be $\frac{\pi}{2}$ or $\cos\phi = 0$. Hence, Equation (2.64) reduces to

$$x_\xi x_\eta + y_\xi y_\eta = 0 \tag{2.65}$$

Equation (2.65) is the *orthogonality condition* to be satisfied at the physical domain boundaries [46].

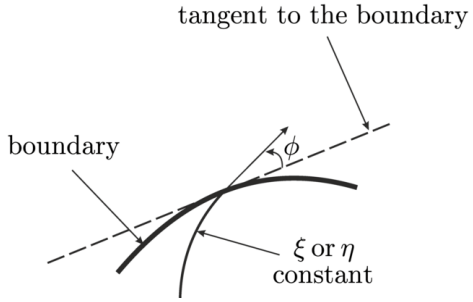


Figure 2-12 The intersection angle ϕ at the boundary of the physical domain.

We now develop the mathematical expressions needed to satisfy the orthogonality condition at the boundaries $S_i (i = 1,2,3,4)$ by evaluating the finite difference discretization of the derivatives appearing in Equation (2.65) at the computational domain boundaries corresponding to the physical domain boundaries where ξ (or η) grid lines should be orthogonal to the boundary. To illustrate the discretization process, consider Figure 2-13 which shows the computational domain boundaries and the finite difference discretization scheme used for the boundary S_1.

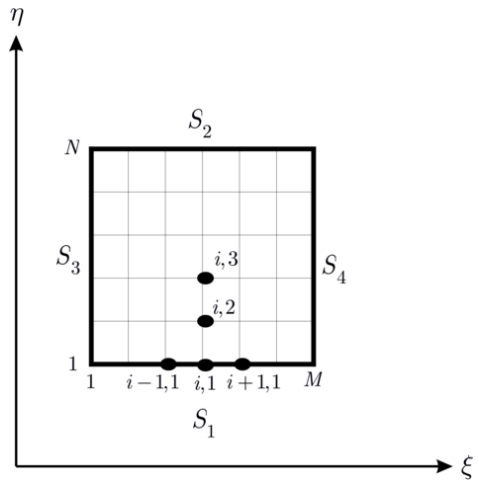


Figure 2-13 The finite difference discretization used for the computational domain boundary S_1.

If Equation (2.65) is satisfied at the boundary S_1, we have

$$x_\xi x_\eta + y_\xi y_\eta = 0 \Rightarrow x_{i,1} \,\&\, y_{i,1} = \text{unknown}$$

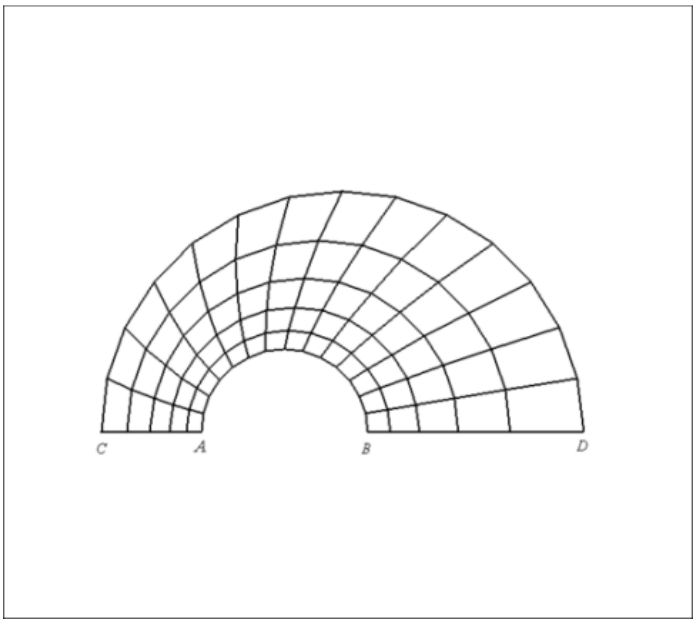
For $x_{i,1}$,

$$\left(\frac{x_{i+1,1} - x_{i-1,1}}{2}\right)\left(\frac{-3x_{i,1} + 4x_{i,2} - x_{i,3}}{2}\right) + \left(\frac{y_{i+1,1} - y_{i-1,1}}{2}\right)\left(\frac{-3y_{i,1} + 4y_{i,2} - y_{i,3}}{2}\right) = 0$$

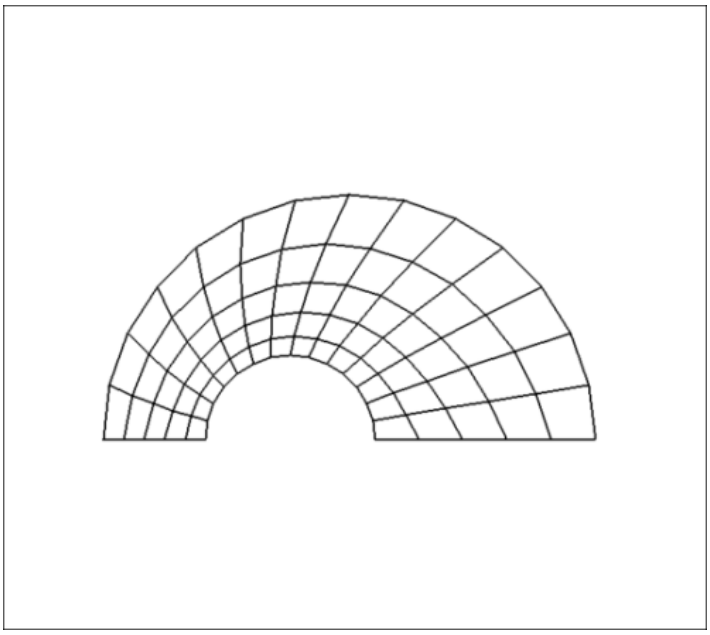
$$x_{i,1} = -\left(-\frac{1}{3}\right)\left[\frac{(y_{i+1,1} - y_{i-1,1})(-3y_{i,1} + 4y_{i,2} - y_{i,3})}{(x_{i+1,1} - x_{i-1,1})} - 4x_{i,2} + x_{i,3}\right] \quad (2.66)$$

The same approach can be used to obtain an expression for $y_{i,1}$ at the boundary S_1 as well as $x_{i,N}$ and $y_{i,N}$ at the boundary S_2, $x_{1,j}$ and $y_{1,j}$ at the boundary S_3, $x_{M,j}$ and $y_{M,j}$ at the boundary S_4. The above relations are valid for $i = 2,\ldots,M-1$ and $j = 2,\ldots,N-1$. Equation (2.66) adjusts the x-position of nodes on the surface S_2, $x_{i,N}(i = 2,\ldots,M-1)$, so that the orthogonality condition, Equation (2.65), is satisfied.

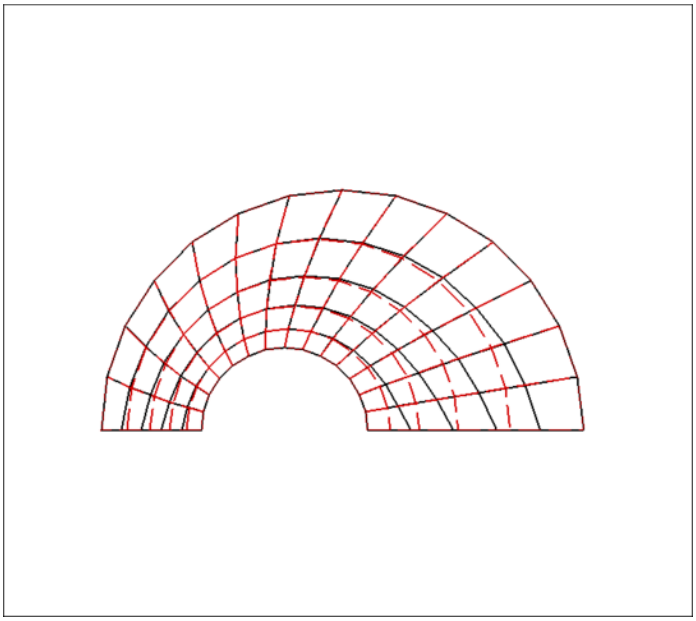
To show the effect of applying the orthogonality condition, consider the body shown in Figure 2-14 meshed with and without applying the orthogonality condition on two faces of the body. As it can be seen from Figure 2-14a, x- position of nodes on the two faces BD and AC are adjusted so that the orthogonality condition is satisfied. In other words, the nodes on the faces BD and AC are no longer uniformly distributed after employing the orthogonality condition.



a) With orthogonality at the boundaries BD and AC.



b) Without orthogonality.



c) Comparison of the grids with and without orthogonality.

Figure 2-14 the effect of orthogonality condition on the grid spacing.

2.7 Validation of the results

So far, all necessary equations for solving a heat transfer problem have been formulated over an irregular domain with Neumann and Robin boundary conditions (both convection and radiation heat transfer modes are allowed). To do this, both algebraic and elliptic grid generation techniques are used to mesh these irregular physical domains and map the domain into a regular computational domain. In addition, the equations required for transforming the heat equation from the physical domain into the computational one have been formulated. Before proceeding any further, the implementation of the equations and the obtained results need to be validated. To do this, recognized commercial finite element softwares such as ANSYS, ABAQUS, or COMSOL can be used. We have used here the ANSYS software. The results from the Fortran codes (based on both TFI and elliptic grid generation) for an arbitrarily chosen set of boundary conditions are compared with the ones from the ANSYS. The boundary conditions are the following (see Figure 2-9):

Boundary conditions:

On surface S_1: heat flux $\dot{q}_1 = 900 \frac{\mathrm{W}}{\mathrm{m}^2}$

On surface S_2: convection heat transfer with $h_2 = 3 \frac{\mathrm{W}}{\mathrm{m}^2.^{\mathrm{o}}\mathrm{C}}$, $T_{\infty_2} = 30^{\circ}\mathrm{C}$

On surface S_3: convection heat transfer with $h_3 = 3 \frac{\mathrm{W}}{\mathrm{m}^2.^{\mathrm{o}}\mathrm{C}}$, $T_{\infty_3} = 30^{\circ}\mathrm{C}$

On surface S_4: convection heat transfer with $h_4 = 3 \frac{\mathrm{W}}{\mathrm{m}^2.^{\mathrm{o}}\mathrm{C}}$, $T_{\infty_4} = 30^{\circ}\mathrm{C}$

Conductivity of physical domain $k = 100 \frac{\mathrm{W}}{\mathrm{m}.^{\mathrm{o}}\mathrm{C}}$

The results are obtained for two different mesh densities of 45×45 and 90×90. The numerical results for maximum and minimum temperatures in the physical domain for the given boundary conditions are reported below. In the following table, the total computation time is the time spent for grid

generation plus the time spent for finding the temperature distribution in the physical domain (direct solver step).

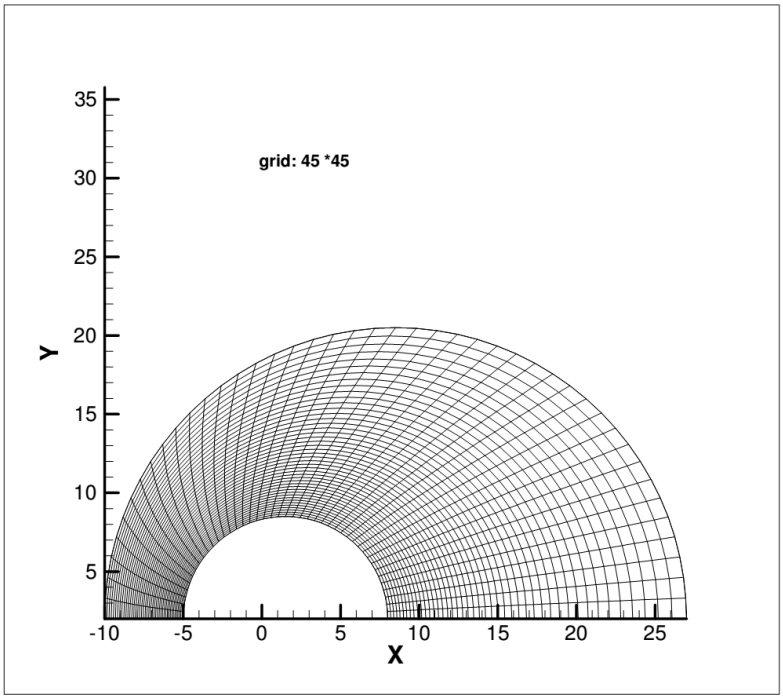
	Grid size	T_{max}(°C)	T_{min}(°C)	Total computation time	
				Grid generation time	Direct solver time
FDM (Elliptic grid generation)	45×45	169.107	70.3637	1 s	29 s
	90×90	169.350	70.4706	3 s	3 min & 7 s
FDM (TFI grid generation)	45×45	169.208	70.3721	<1 s	19 s
	90×90	169.348	70.4636	1 s	3 min & 3 s
FEM (ANSYS)	~ 8100 elements	169.434	70.508	18 s	17 s

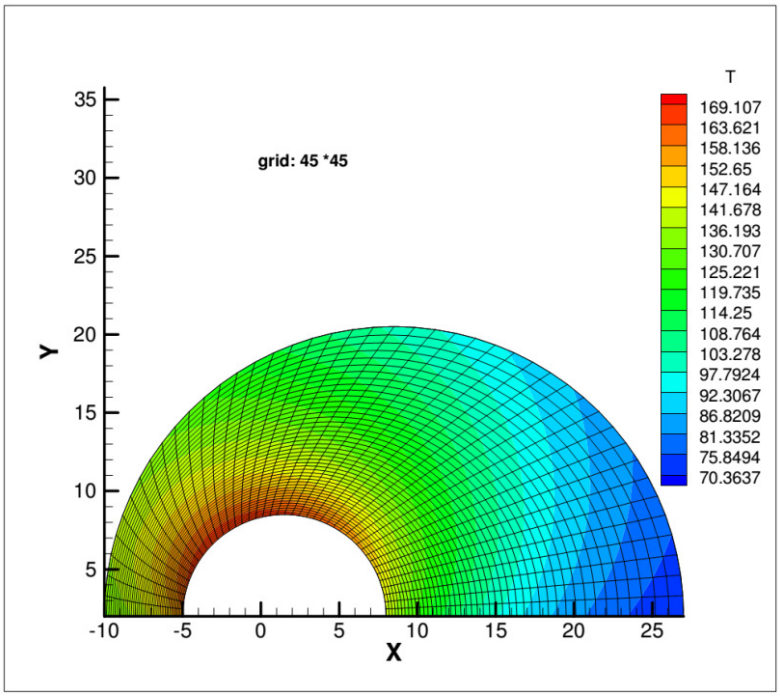
Table 2-1 comparison of the results from FDM (using TFI and Elliptic grid generations) and FEM package ANSYS.

Table 2-1 shows the close agreement between the maximum and minimum temperatures in the given physical domains obtained from TFI and elliptic grid generation techniques and the finite element package ANSYS. A tolerance 10^{-9} is used in the iteration loop to reduce the round-off error and to ensure adequate accuracy for results. From the results in Table 2-1 we conclude that the computation time for grid generation using both elliptic and TFI methods are less than the one using the unstructured grid generation in ANSYS. However, the direct solver in ANSYS obtains the results sooner than the direct solver in FDM for the same grid size. The reason is the iterative nature of FDM. An iterative process (SOR) is used in FDM (both using the elliptic and TFI grid generation methods) to obtain the temperature distribution. In our shape optimization problems, however, this relatively high computation time for the direct solver in FDM is for the first iteration of optimization only because the initial guess is far from the solution. As will be shown, the subsequent iterations in optimization process require much less computation time than does the first iteration because the solution at any iteration is an initial guess for the next iteration. In other words, the second iteration initial guess (first iteration solution) is very close to the second iteration solution, and so forth. Hence the direct solver converges very fast in

subsequent iterations. The results and the meshed physical domains and their associated temperature distributions are depicted below. Two different grid sizes are used to mesh the physical domain and then to obtain the temperature distribution using TFI and elliptic grid generation methods.

***Elliptic grid generation*:**





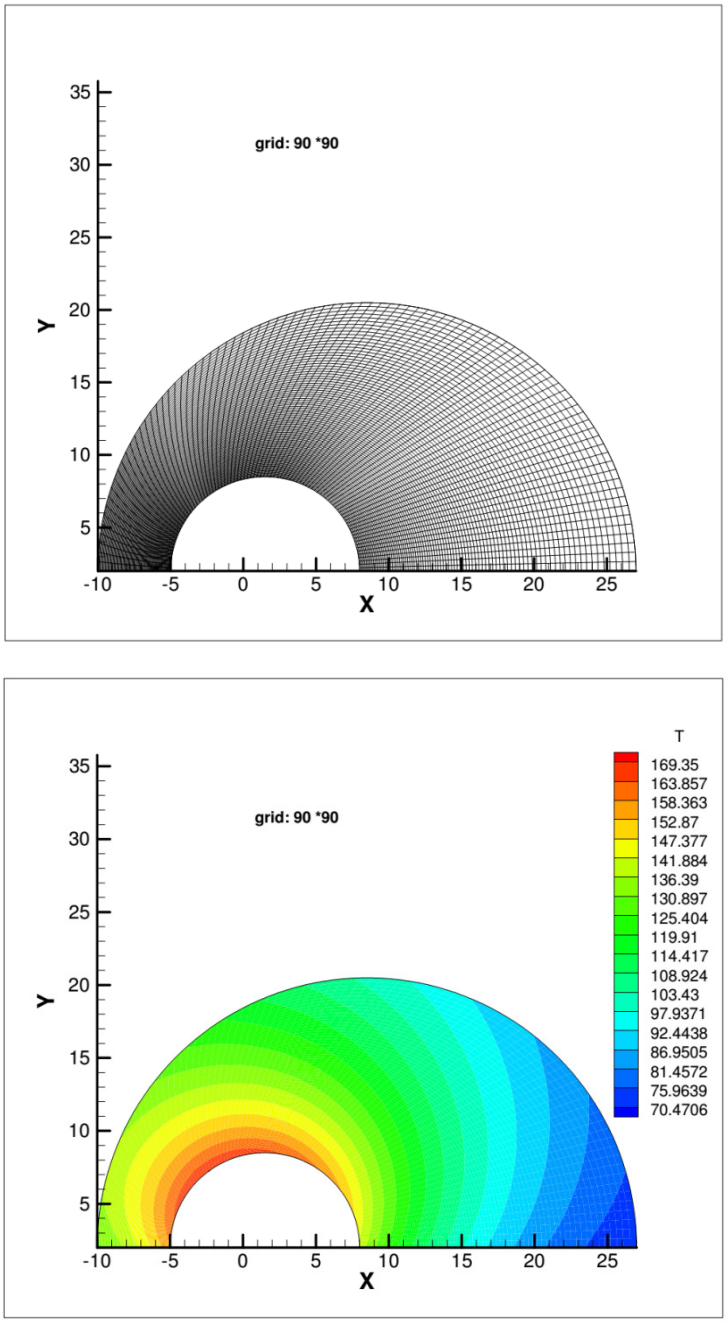
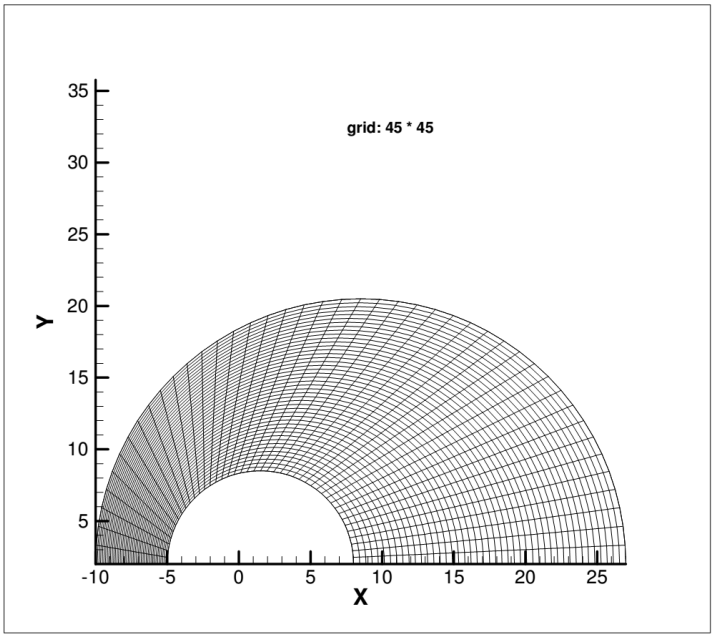
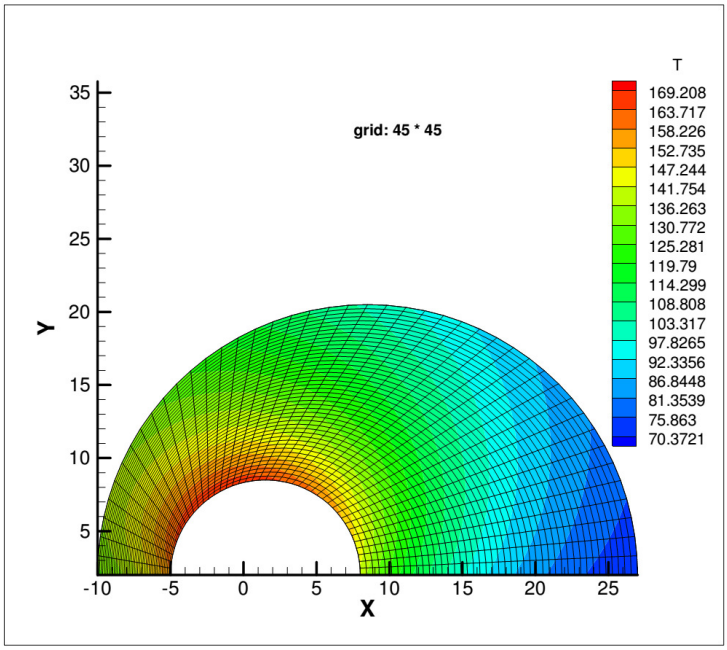


Figure 2-15 The results from application of the elliptic grid generation method in solving a steady state heat conduction problem to find the temperature distribution in an irregular body.

***Algebraic grid generation*:**





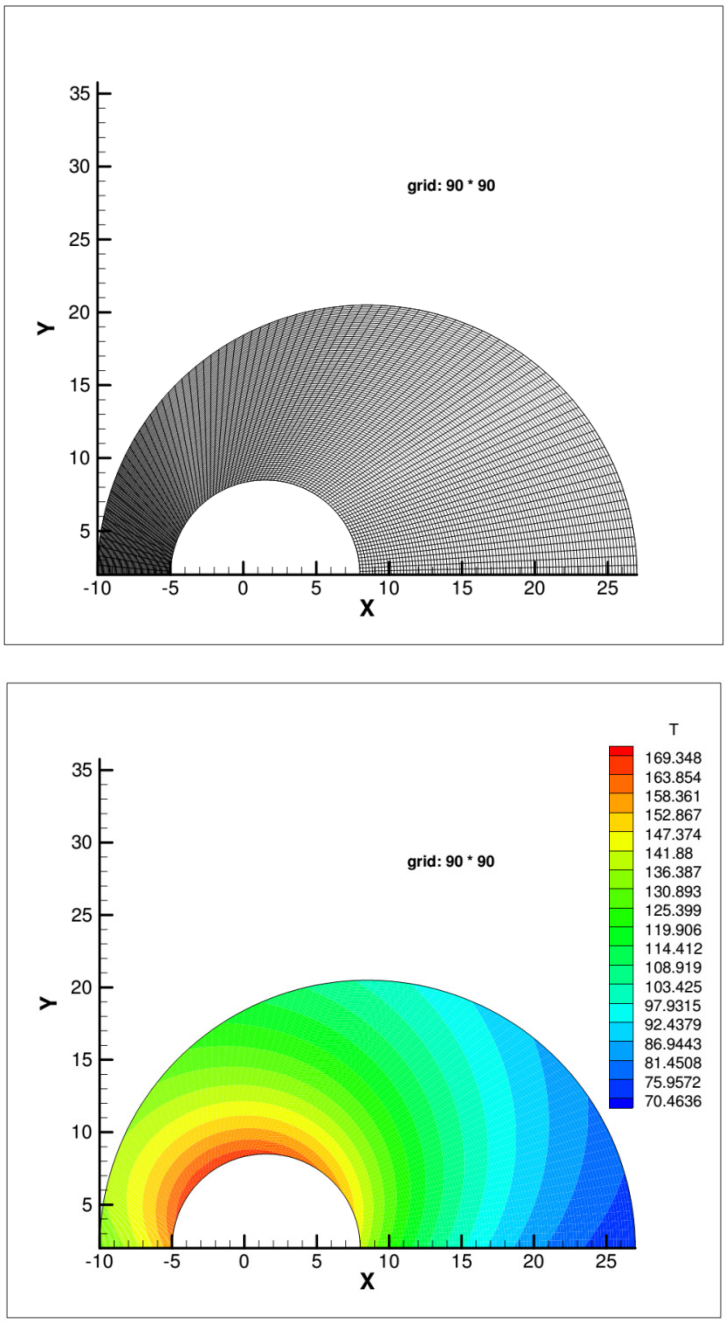
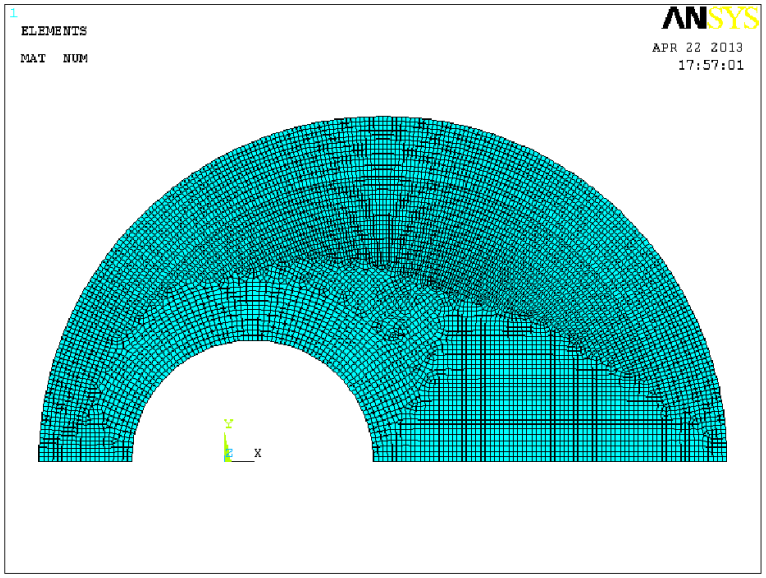
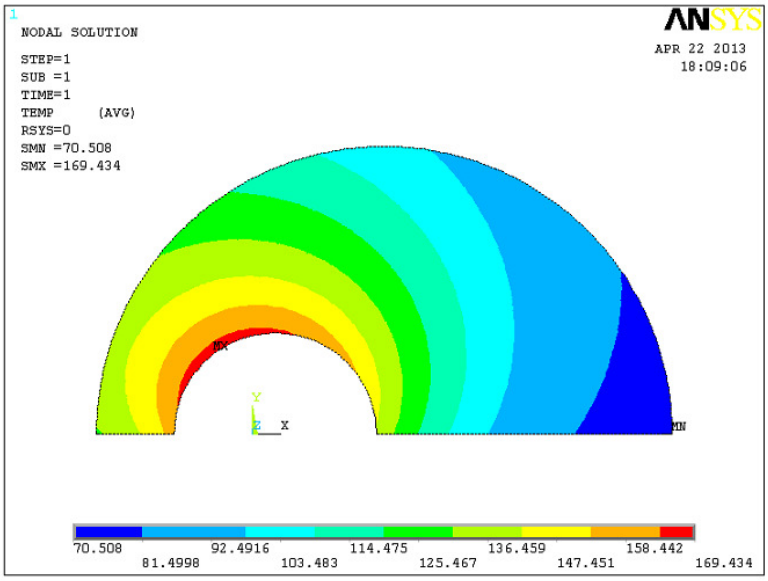


Figure 2-16 The results from application of the TFI grid generation method in solving a steady state heat conduction problem to find the temperature distribution in an irregular body.

The results obtained from ANSYS (V.13) for the same physical domain and same boundary conditions are shown below.



a) Grid Generation



b) Temperature profile

Figure 2-17 ANSYS simulation results (a) mesh; (b) temperature contours.

A second example with different boundary conditions is now considered to compare FDM and FEM. Unlike the first example in which $h_2 = h_3 = h_4$ and $T_{\infty_2} = T_{\infty_3} = T_{\infty_4}$, these values are different in second example. The applied boundary conditions are now the following:

Boundary conditions:

On surface S_1: heat flux $\dot{q}_1 = 700 \frac{\mathrm{W}}{\mathrm{m}^2}$

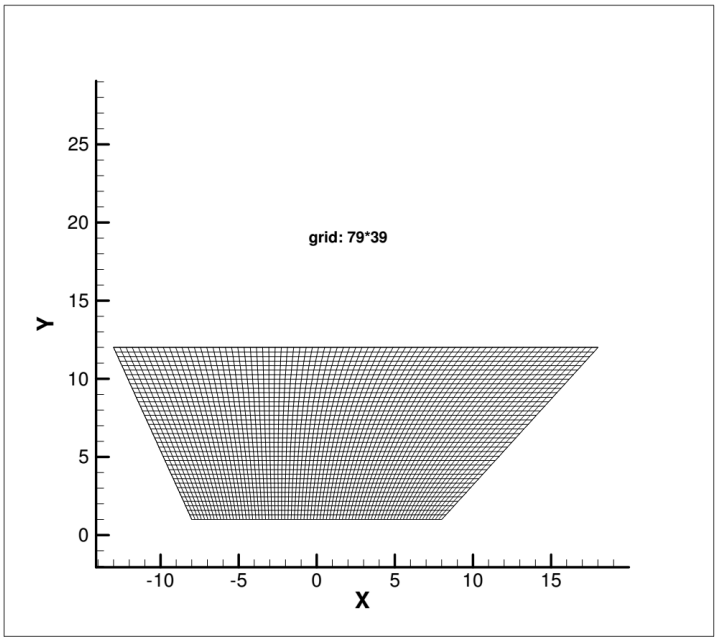
On surface S_2: convection heat transfer with $h_2 = 6 \frac{\mathrm{W}}{\mathrm{m}^2.^{\mathrm{o}}\mathrm{C}}$, $T_{\infty_2} = 30^{\circ}\mathrm{C}$

On surface S_3: convection heat transfer with $h_3 = 8 \frac{\mathrm{W}}{\mathrm{m}^2.^{\mathrm{o}}\mathrm{C}}$, $T_{\infty_3} = 35^{\circ}\mathrm{C}$

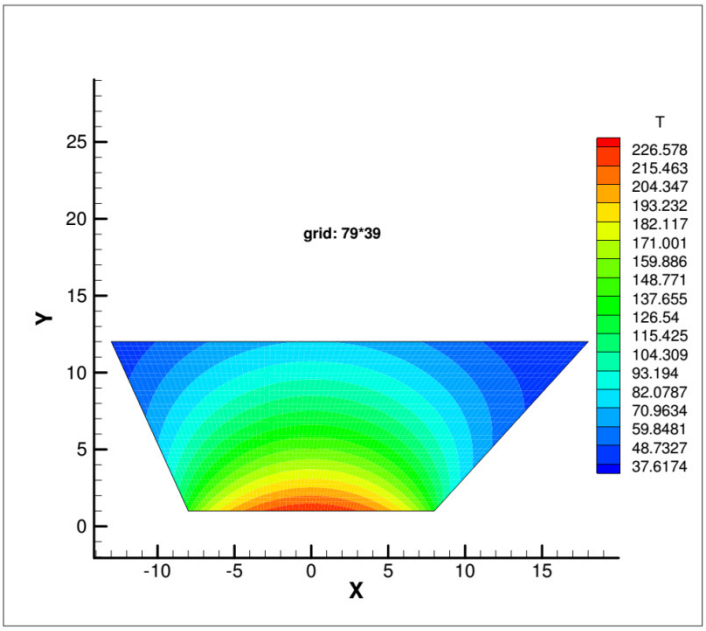
On surface S_4: convection heat transfer with $h_4 = 10 \frac{\mathrm{W}}{\mathrm{m}^2.^{\mathrm{o}}\mathrm{C}}$, $T_{\infty_4} = 40^{\circ}\mathrm{C}$

Conductivity of physical domain $k = 30 \frac{\mathrm{W}}{\mathrm{m}.^{\mathrm{o}}\mathrm{C}}$

The temperature distribution obtained with FDM (elliptic and TFI grid generation techniques) and ANSYS can be seen in Figure 2-18 through Figure 2-20, respectively. In addition, Table 2-2 reports the maximum and minimum temperatures computed by ANSYS or using FDM. A tolerance 10^{-9} is used for the iteration loop in FDM for both grid generation and the direct solver when using the elliptic grid generation method and for the direct solver when using the TFI grid generation method (the TFI grid generation is an algebraic method and does not need the iteration).

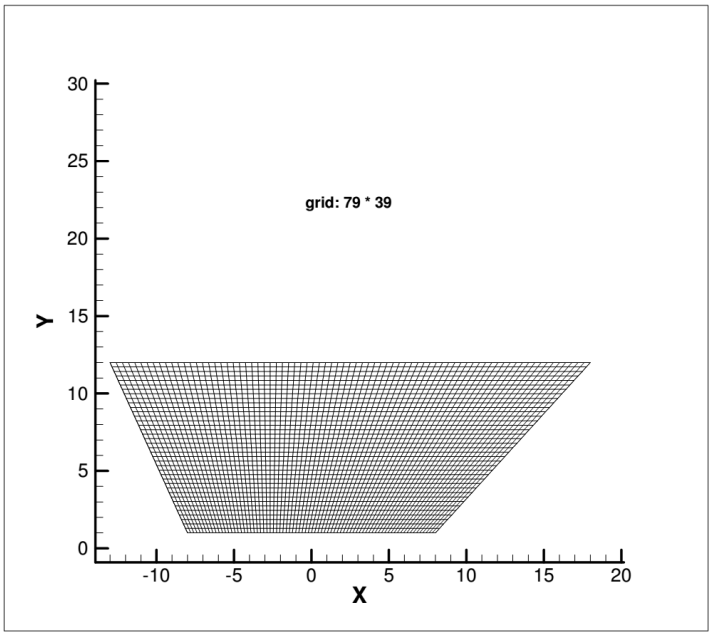


a) Grid

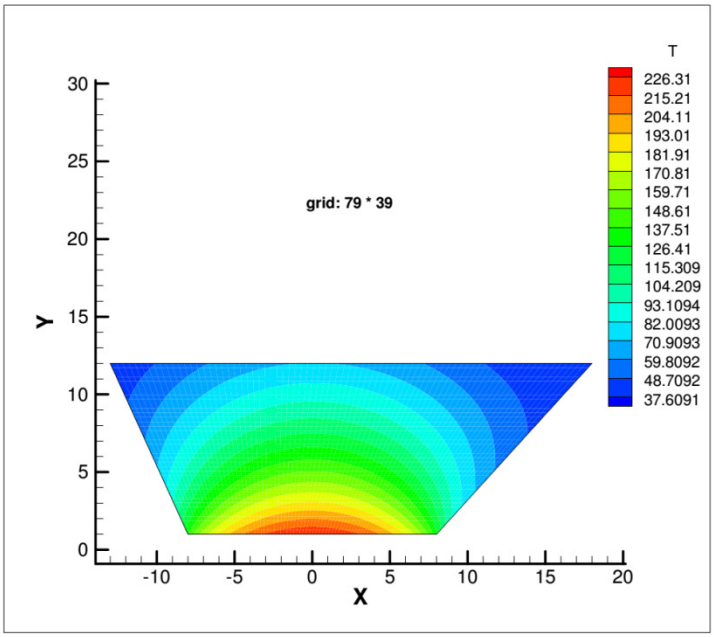


b) Temperature profile

Figure 2-18 The grid (a) and the temperature distribution (b) from the elliptic grid generation technique.

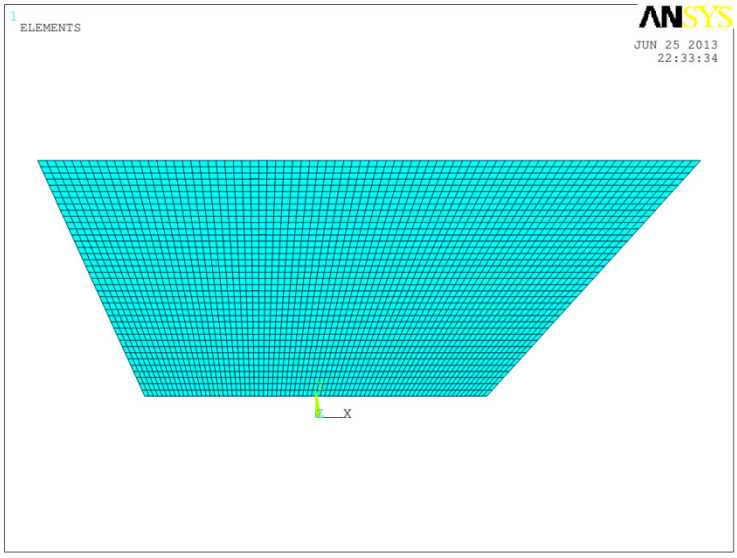


a) Grid

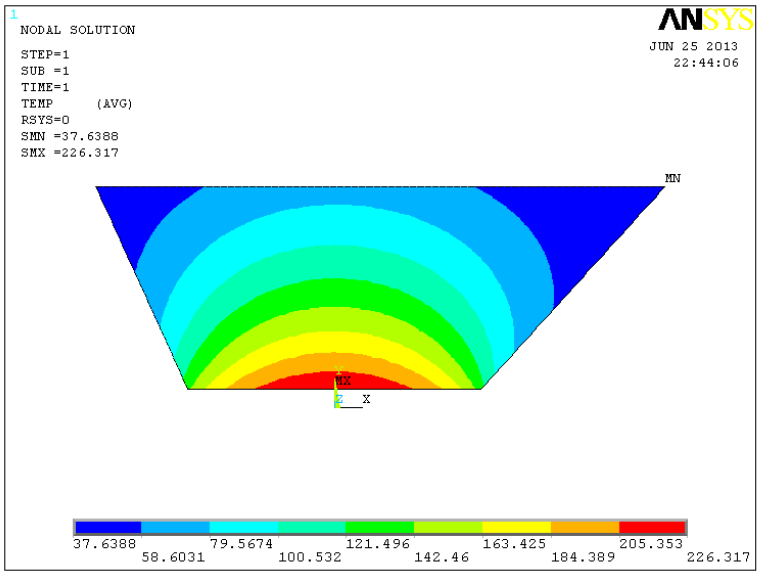


b) Temperature profile

Figure 2-19 The grid (a) and the temperature distribution (b) from the TFI grid generation technique.



a) Grid Generation



b) Temperature contours

Figure 2-20 ANSYS simulation results (a) mesh; (b) temperature contours.

	Grid size	$T_{max}(°C)$	$T_{min}(°C)$	Total computation time	
				Grid generation time	Direct solver time
FDM (Elliptic grid generation)	79×39	226.578	37.6174	1 s	19 s
FDM (TFI grid generation)	79×39	226.31	37.6091	1 s	40 s
FEM (ANSYS)	79×39	226.317	37.6388	3 s	6 s

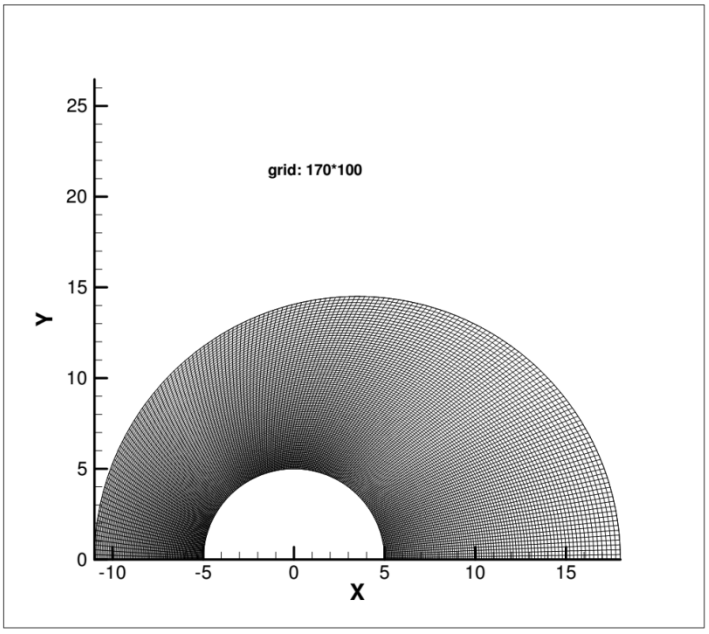
Table 2-2 the comparison of FDM (TFI and Elliptic grid generation methods) and FEM (ANSYS)

The data for the validation of a heat transfer problem test case with the radiation boundary conditions (Equation (2.55)) are as follows

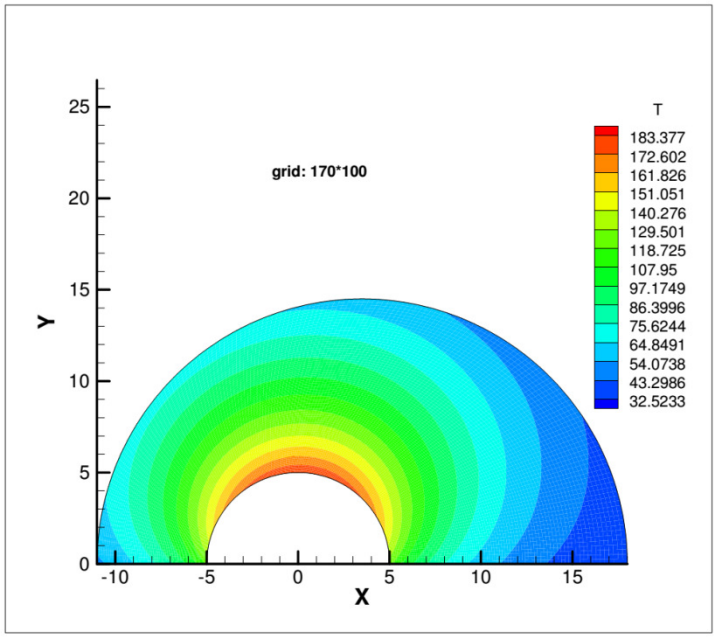
$$k = 30(\frac{\mathrm{W}}{\mathrm{m}.^{\circ}\mathrm{C}}), T_{S_i}(i = 2,3,4) = 20^{\circ}\mathrm{C}, \sigma = 5.6704 \times 10^{-8}(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{K}^4}), \dot{q} = 800(\frac{\mathrm{W}}{\mathrm{m}^2})$$

$$\varepsilon_i(i = 2,3,4) = 0.7$$

The results from both the elliptic grid generation technique and ANSYS and the body geometric configuration are depicted below

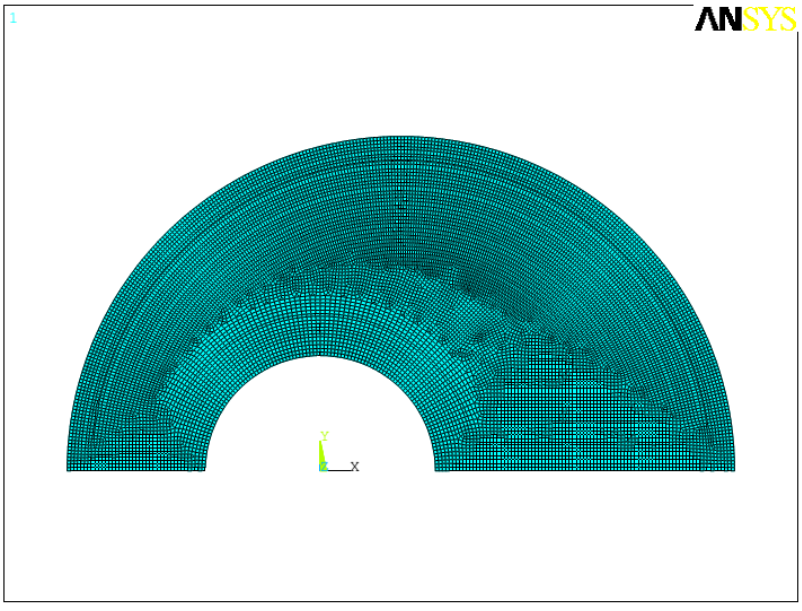


a) Grid generation

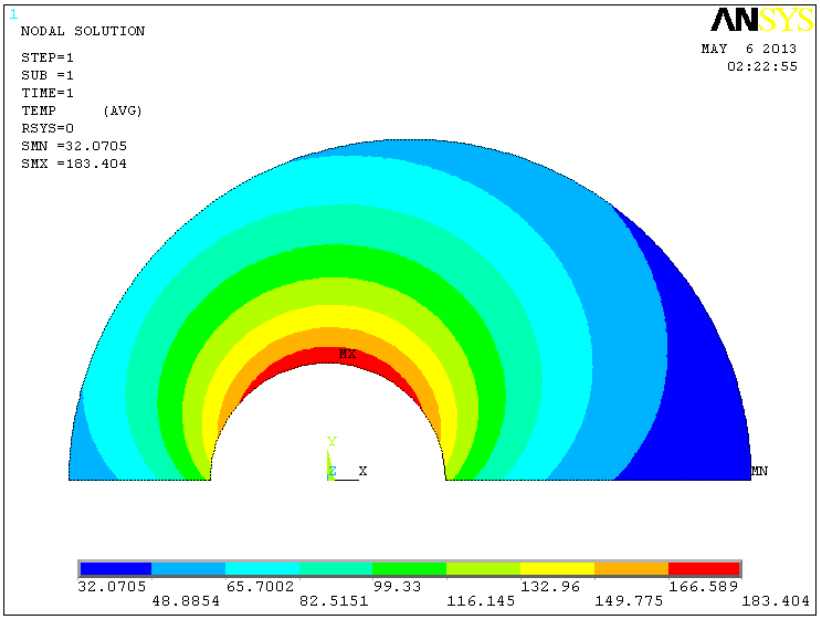


b) Temperature distribution

Figure 2-21 The grid (a) and the temperature distribution (b) from the elliptic grid generation technique.



a) Grid generation



b) Temperature contour

Figure 2-22 ANSYS simulation results (a) mesh; (b) temperature contours.

As it can be seen from the above figures, the maximum and minimum values for the temperatures obtained from using both the elliptic grid generation method and ANSYS are comparable. The values are given below

	Grid size	T_{max}(°C)	T_{min}(°C)	Total computation time	
				Grid generation time	Direct solver time
FDM(Elliptic grid generation)	170×100	183.377	32.5233	7s	3min & 55s
FEM (ANSYS)	~ 17000 elements	183.404	32.0705	16s	16s

Table 2-3 The comparison of the results from using FDM (elliptic grid generation) and FEM (ANSYS) for a heat conduction problem with radiation boundary conditions.

Therefore, the *percent difference* of the minimum and maximum temperatures resulting from the procedure based on body-fitted (elliptic) grid generation (BFGG) and the ANSYS may be obtained from the following formulas:

$$\%\text{Diff. in } T_{\max} = \left| \frac{(T_{\max})_{\text{BFGG}} - (T_{\max})_{\text{ANSYS}}}{(\frac{(T_{\max})_{\text{BFGG}} + (T_{\max})_{\text{ANSYS}}}{2})} \right| \times 100 = \%0.0147 \quad (2.67)$$

$$\%\text{Diff. in } T_{\min} = \left| \frac{(T_{\min})_{\text{BFGG}} - (T_{\min})_{\text{ANSYS}}}{(\frac{(T_{\min})_{\text{BFGG}} + (T_{\min})_{\text{ANSYS}}}{2})} \right| \times 100 = \%1.388 \quad (2.68)$$

The Fortran codes are compiled using a PGI compiler and computations are run on a PC with Intel Pentium Dual 1.73 and 1 G RAM.

Straightforward implementation and accuracy of results are two main factors in choosing an appropriate method to proceed with our shape optimization problem. The comparison of results obtained from the use of the proposed approach based on the elliptic grid generation technique and the direct solver for the heat conduction equation and ones obtained from ANSYS (FEM) reveals a good agreement between the results thereby confirming the correct implementation of the proposed method. As mentioned before, the relatively high computation time in FDM is for the first iteration of optimization only and the direct solver in FDM converges very fast in subsequent iterations. As will be shown in the following chapter, the sensitivity analysis will be performed using a novel method based on FDM.

2.8 Conclusion

This chapter has confirmed the correct implementation of the grid generation techniques and the solution procedure for the steady heat conduction equation. The proposed approach is based on mapping the irregular physical domain into a regular computational domain and takes advantage of the discretization of the governing equation using the FDM. The FDM can be easily implemented and is not computationally intensive. Furthermore,

comparison of the results obtained from the approach and the ones from FEM (ANSYS) revealed high accuracy of the proposed approach. As will be shown in the following chapter, however, the main advantage of this solution procedure is the easy incorporation of the sensitivity analysis into the optimal shape design algorithm. Indeed mapping the physical domain into a fixed computational domain provides for a very efficient computation of the sensitivity matrix.

3. Two dimensional sensitivity analysis and shape optimization

3.1 Introduction

Before dealing with the concept of the shape optimization, it is necessary to define several terms encountered in this field.
In the context of the optimization, the aim is to minimize one or several criteria. These criteria are completely dependent on the problem under consideration. In other words, in any field of study, the objective function(s) has its own definition. For instance, in aerodynamics, the objective function may be a measure of the drag force exerted by the fluid on a body and the optimal shape design problem may be to find the shape which minimizes the drag.
The mathematical representation of the heat conduction problem in the shape optimization problem of interest here can be expressed as below (see Figure 1-2)

$$\nabla^2 T = 0 \qquad \text{in physical domain } \Omega \tag{3.1}$$

subject to the boundary conditions

$$\frac{\partial T}{\partial n} = -\frac{\dot{q}}{k} \text{ on boundary surface } \Gamma_1 \tag{3.2}$$

$$\frac{\partial T}{\partial n} = -\frac{h}{k}(T_{\Gamma_i} - T_{\infty_i}) \text{ on boundary surface } \Gamma_i, i=2,3,4 \tag{3.3}$$

The aim of shape optimization problem is to design the outer surface Γ_2 so that the square of the difference between the outer surface temperature and a desired outer temperature distribution is minimized. In more formal mathematical terms, the shape optimization problem can be represented as

$$\min_{(x,y)\in\Gamma_2}\left\{\mathcal{J}(x,y) := \| T_{(x,y)} - T_d \|^2 : \nabla^2 T = 0 \text{ in } \Omega \text{ , BCs in Eqs.(3.2 \& 3.3)}\right\} \quad (3.4)$$

where x and y are coordinates of nodes on a boundary, respectively. In the problem of interest, the design parameters x and y define the shape of the outer surface Γ_2. Let $x_{i,N}^{(k)}$ and $y_{i,N}^{(k)} (i = 1,2,3,\ldots,M)$ be the coordinates of the node $F_{i,N}$ on the outer boundary to be optimized at iteration k (see Figure 3-1). The locations of the nodes on this outer surface have to be numerically optimized for the objective function

$$\mathcal{J} = \sum_{i=1}^{M} (T_{i,N} - T_d)^2 \quad (3.5)$$

to reach a minimum. $T_{i,N}$ and T_d are the nodal temperature at point $F_{i,N} (i = 1,2,3,\ldots,M)$ and the desired outer surface temperature, respectively. The objective function is naturally dependent on the location of the nodes on the outer surface since the locations of the nodes define the shape of the body under consideration at iteration k. The aim of the shape optimization problem is to find a set of design parameters (outer surface node location) under given boundary conditions which minimize the objective function mentioned above.

3.2 Sensitivity analysis

In gradient based shape optimization, it is required to calculate the effect of the variation of the shape on the objective function. The computation of derivative of the objective function with respect to the shape (design parameters) is an integrated part of gradient based shape optimization which is called the sensitivity analysis. In other words, the computation of the sensitivity of the objective function to the shape is called the *sensitivity analysis* or shape sensitivity. In the following, we will restrict our study to the heat transfer problem described above, introduce the sensitivity matrix (*Jacobian matrix*), and investigate methods for calculating the sensitivity coefficients or components of the Jacobian matrix Ja. As shown in Figure

3-1, suppose that we wish to calculate the sensitivity of the objective function defined by Equation (3.1) to the locations of nodes on the surface $S_2\left(CD\right)$. Side AB is subject to a prescribed heat flux $\dot{q}$ and sides AC, CD, and DB to convective or radiative heat transfer.

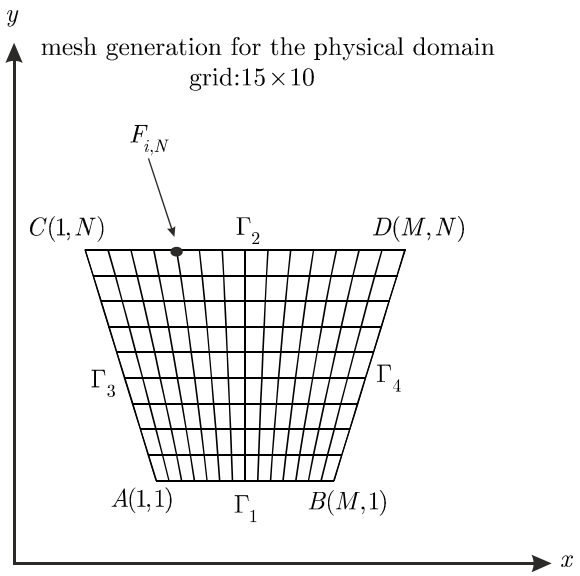


Figure 3-1 Illustration of the computational domain, associated mesh, and notations. The location of the nodes on the outer upper surface Γ_2 has to be numerically optimized for the objective function value to reach a minimum.

There are several different methods to perform the sensitivity analysis. We introduce briefly below three such approaches, including (i) the direct analytic solution, (ii) the boundary value problem, and (iii) the perturbation approach [25].

The direct analytic solution:

If there is an analytical solution for the temperature field in a direct linear heat transfer problem, then the sensitivity coefficients with respect to an unknown parameter P can be determined by differentiating the solution with respect to P [25]. For example, for a long solid cylinder of radius r_0, thermal conductivity k, cylinder side surface constant temperature T_S, and heat

generation rate $\dot{g}$, the variation of the temperature in the cylinder is given by [52]

$$T(r) = \frac{\dot{g} r_0^2}{k}\left[1 - (\frac{r}{r_0})^2\right] + T_S$$

Then the sensitivity coefficient with respect to, say, heat generation rate $\dot{g}$ is given by

$$Ja_{\dot{g}} = \frac{\partial T(r)}{\partial \dot{g}} = \frac{r_0^2}{k}\left[1 - (\frac{r}{r_0})^2\right]$$

which is independent of the heat generation rate $\dot{g}$. Then, the inverse problem of estimating $\dot{g}$ is linear. The sensitivity coefficients with respect to each variable involved in the above relationship may be determined in a similar way. It can be shown that the inverse problem of estimating k is nonlinear.

The boundary value problem:

In this method, a boundary value problem can be formulated to find the sensitivity coefficients by differentiating the original direct heat transfer problem with respect to the unknown coefficients. For example, consider the following heat conduction problem

$$k\frac{\partial^2 T}{\partial x^2} = C\frac{\partial T}{\partial t} \qquad \text{in } 0 < x < L \qquad \text{for } t > 0 \qquad \text{(a)}$$

$$-k\frac{\partial T}{\partial x} = \dot{q}_0 \qquad \text{at } \quad x = 0 \qquad \text{for } t > 0 \qquad \text{(b)}$$

$$\frac{\partial T}{\partial x} = 0 \qquad \text{at } \quad x = L \qquad \text{for } t > 0 \qquad \text{(c)}$$

$$T = T_i \qquad \text{in } \ 0 < x < L \qquad \text{for } \ t = 0 \qquad \text{(d)}$$

where $C = \rho c_p$ is heat capacity, $\dot{q}_0$ is applied heat flux, and k is thermal conductivity. Suppose we want to construct the sensitivity problem for determining the sensitivity coefficients with respect to thermal conductivity.

Denoting the sensitivity coefficients with respect to k by $Ja_k \equiv \dfrac{\partial T}{\partial k}$ and differentiating the above equations with respect to k, we obtain the following boundary value problem for determination of the sensitivity coefficients

$$k\frac{\partial^2 Ja_k}{\partial x^2} + \frac{\partial^2 T}{\partial x^2} = C\frac{\partial Ja_k}{\partial t} \qquad \text{in } 0 < x < L \qquad \text{for } t > 0 \qquad (\text{a}')$$

$$\frac{\partial T}{\partial x} + k\frac{\partial Ja_k}{\partial x} = 0 \qquad \text{at} \quad x = 0 \qquad \text{for } t > 0 \qquad (\text{b}')$$

$$\frac{\partial Ja_k}{\partial x} = 0 \qquad \text{at} \quad x = L \qquad \text{for } t > 0 \qquad (\text{c}')$$

$$Ja_k = 0 \qquad \text{in} \quad 0 < x < L \qquad \text{for} \quad t = 0 \qquad (\text{d}')$$

For example, the derivation of Equation (b') is as follows

$$\frac{\partial}{\partial k}(-k\frac{\partial T}{\partial x}) = \frac{\partial}{\partial k}\dot{q}_0$$

$$\left[\frac{\partial}{\partial k}(-k)\right]\frac{\partial T}{\partial x} + \left[\frac{\partial}{\partial k}(\frac{\partial T}{\partial x})\right](-k) = \frac{\partial}{\partial k}\dot{q}$$

$$-1(\frac{\partial T}{\partial x}) - k\frac{\partial^2 T}{\partial k \partial x} = 0$$

and multiplying both sides by (-1)

$$\frac{\partial T}{\partial x} + k\frac{\partial Ja_k}{\partial x} = 0$$

Equations (a') and (b') contain the non-homogeneous terms $\dfrac{\partial^2 T}{\partial x^2}$ and $\dfrac{\partial T}{\partial x}$, respectively. Furthermore, the unknown parameter k appears in these two equations; thus, the problem of estimating k is nonlinear. The solution of the above equations yields the sensitivity coefficients Ja_k [25].

The perturbation approach:

Suppose we wish to calculate the sensitivity of temperature of nodes on the boundary Γ_2 (see Figure 3-1), $T_{i,N}(i=1,\ldots,M)$, to the $x-$position of the nodes on the boundary Γ_4, $x_{M,j}(j=1,\ldots,N)$. The sensitivity analysis can be performed by introducing small perturbations to the $x-$coordinate of each point on the boundary Γ_4, $x_{M,j}$, individually. The grid generation and direct problem may be solved for this perturbated shape to obtain the new temperatures $T_{i,N}$. Using these values for the temperatures, one can evaluate the dependency of the temperatures $T_{i,N}$ to the perturbation of the $x-$position of points of coordinates $(M,j), j=1,\ldots,N$. The finite difference method may be used to formulate these perturbations as follows

$$\frac{\partial T_{i,N}}{\partial x_{M,j}} = \frac{T_{i,N}(x_{M,j}+\varepsilon x_{M,j}) - T_{i,N}(x_{M,j})}{\varepsilon x_{M,j}}$$

where ε can be, say, 10^{-6}. The term $\varepsilon x_{M,j}$ is the perturbation in the $x-$position of points of coordinates $(M,j), j=1,\ldots,N$. Since the sensitivity of each temperature $T_{i,N}(i=1,\ldots,M)$ to each $x-$position of points of coordinates $(M,j), j=1,\ldots,N$ is required, the computation of the sensitivity coefficients using this method requires N additional solution of direct heat transfer problem. Hence, this method is only suitable when the number of points on the boundary Γ_4, N, is small. Thus, for the shape optimization problems in heat transfer, which demand a fine grid to obtain accurate results, the perturbation method using the finite difference method is a very time consuming and error-prone process. Furthermore, introducing the $y-$coordinate increases the complexity of the process. In this study, we will introduce a novel method to calculate the sensitivity coefficients based on the finite difference method and transformation rules introduced in Chapter 2. As will be shown, it requires only one solution of the direct problem (at each iteration) to compute all sensitivity coefficients. Therefore, as far as the computation time and the computer resources are concerned, it is a much preferable method to perform the sensitivity analysis.

Adjoint method:

The adjoint method is an alternative approach to compute the gradient vector. In this method, an extended objective function expression is obtained by multiplying the partial differential equation of the direct problem (Laplace equation, in this case) by a Lagrange multiplier, λ, and adding the resulting expression to the original objective function (Equation (3.5)). This method is very efficient because the gradient vector is computed with the solution of a single adjoint problem [25, 27, 86]. The analytical derivation of the adjoint method and the computation of the gradient vector using this method are not in the scope of the thesis.

A novel method based on FDM:

In this method, we introduce a novel method based on FDM which computes all sensitivity coefficients in only one solution of the direct problem (at each iteration). Therefore, unlike the perturbation method, this novel method is very efficient for problems involving large number of variables (node positions).

With regard to Equation (3.5), the following equations can be written in order to calculate the Jacobian matrix

$$\frac{\partial \boldsymbol{\mathcal{J}}}{\partial x_{l,N}} = 2\sum_{i=1}^{M}(T_{i,N} - T_d)\frac{\partial T_{i,N}}{\partial x_{l,N}} \tag{3.6}$$

$$\frac{\partial \boldsymbol{\mathcal{J}}}{\partial y_{l,N}} = 2\sum_{i=1}^{M}(T_{i,N} - T_d)\frac{\partial T_{i,N}}{\partial y_{l,N}} \tag{3.7}$$

where $i \in \left[1, M\right]$ and $l \in \left[1, M\right]$. The Jacobian matrix coefficients for this problem are defined as following

$$Ja_{x_{i,l}} = \frac{\partial T_{i,N}}{\partial x_{l,N}} \tag{3.8}$$

$$Ja_{y_{i,l}} = \frac{\partial T_{i,N}}{\partial y_{l,N}} \tag{3.9}$$

If the sensitivity of the objective function defined by Equation (3.5) to the profile of side, say, AC (see Figure 3-1) is required, they can be written as

$$Ja_{x_{i,j}} = \frac{\partial T_{i,N}}{\partial x_{1,j}} \tag{3.10}$$

$$Ja_{y_{i,j}} = \frac{\partial T_{i,N}}{\partial y_{1,j}} \tag{3.11}$$

where $i \in \left[1, M\right]$ and $j \in \left[1, N\right]$.

Alternatively, if the sensitivity of the objective function to the profile of side DB is required, they can be written as

$$Ja_{x_{i,j}} = \frac{\partial T_{i,N}}{\partial x_{M,j}} \tag{3.12}$$

$$Ja_{y_{i,j}} = \frac{\partial T_{i,N}}{\partial y_{M,j}} \tag{3.13}$$

In explicit form, the sensitivity matrices for Equations (3.8) and (3.9), respectively, can be written as

$$\mathbf{Ja}_x = \begin{bmatrix} \frac{\partial T_{1,N}}{\partial x_{1,N}} & \frac{\partial T_{1,N}}{\partial x_{2,N}} & \frac{\partial T_{1,N}}{\partial x_{3,N}} & \cdots & \frac{\partial T_{1,N}}{\partial x_{M,N}} \\ \frac{\partial T_{2,N}}{\partial x_{1,N}} & \frac{\partial T_{2,N}}{\partial x_{2,N}} & \frac{\partial T_{2,N}}{\partial x_{3,N}} & \cdots & \frac{\partial T_{2,N}}{\partial x_{M,N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial T_{M,N}}{\partial x_{1,N}} & \frac{\partial T_{M,N}}{\partial x_{2,N}} & \frac{\partial T_{M,N}}{\partial x_{3,N}} & \cdots & \frac{\partial T_{M,N}}{\partial x_{M,N}} \end{bmatrix} \tag{3.14}$$

$$\mathbf{Ja}_{y} = \begin{bmatrix} \dfrac{\partial T_{1,N}}{\partial y_{1,N}} & \dfrac{\partial T_{1,N}}{\partial y_{2,N}} & \dfrac{\partial T_{1,N}}{\partial y_{3,N}} & \cdots & \dfrac{\partial T_{1,N}}{\partial y_{M,N}} \\ \dfrac{\partial T_{2,N}}{\partial y_{1,N}} & \dfrac{\partial T_{2,N}}{\partial y_{2,N}} & \dfrac{\partial T_{2,N}}{\partial y_{3,N}} & \cdots & \dfrac{\partial T_{2,N}}{\partial y_{M,N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial T_{M,N}}{\partial y_{1,N}} & \dfrac{\partial T_{M,N}}{\partial y_{2,N}} & \dfrac{\partial T_{M,N}}{\partial y_{3,N}} & \cdots & \dfrac{\partial T_{M,N}}{\partial y_{M,N}} \end{bmatrix} \tag{3.15}$$

Since the physical domain is mapped onto the computational one, the chain rule may be used to correlate variables in two domains. Accordingly,

$$\frac{\partial T_{i,N}}{\partial x_{l,N}} = \frac{\partial T_{i,N}}{\partial \xi}\frac{\partial \xi}{\partial x_{l,N}} + \frac{\partial T_{i,N}}{\partial \eta}\frac{\partial \eta}{\partial x_{l,N}} \tag{3.16}$$

$$\frac{\partial T_{i,N}}{\partial y_{l,N}} = \frac{\partial T_{i,N}}{\partial \xi}\frac{\partial \xi}{\partial y_{l,N}} + \frac{\partial T_{i,N}}{\partial \eta}\frac{\partial \eta}{\partial y_{l,N}} \tag{3.17}$$

By interchanging x and ξ, and y and η, and solving the derived equations for $\dfrac{\partial T}{\partial x}$ and $\dfrac{\partial T}{\partial y}$, we finally obtain (see Equations (2.7) through (2.12) for more details)

$$\frac{\partial T_{i,N}}{\partial x_{l,N}} = \frac{1}{J}[(y_{\eta})_{l,N}(T_{\xi})_{i,N} - (y_{\xi})_{l,N}(T_{\eta})_{i,N}] \tag{3.18}$$

$$\frac{\partial T_{i,N}}{\partial y_{l,N}} = \frac{1}{J}[-(x_{\eta})_{l,N}(T_{\xi})_{i,N} + (x_{\xi})_{l,N}(T_{\eta})_{i,N}] \tag{3.19}$$

where $J = (x_{\xi}y_{\eta} - x_{\eta}y_{\xi})_{l,N}$ is the Jacobian of the transformation. Now using the finite difference method to discretize the equations in the computational domain we can write appropriate algebraic approximations for all partial derivatives involved in the above equations. For example,

$$(T_{\xi})_{i,N} = \frac{T_{i+1,N} - T_{i-1,N}}{2} \tag{3.20}$$

$$(T_{\eta})_{i,N} = \frac{3T_{i,N} - 4T_{i,N-1} + T_{i,N-2}}{2} \tag{3.21}$$

The same procedure can be followed to find the relations for $(x_{\eta})_{l,N}$,$(y_{\eta})_{l,N}$, $(x_{\xi})_{l,N}$, and $(y_{\xi})_{l,N}$. These formulae allow the direct evaluation of the shape sensitivities using the computed temperature field.

There are several procedures to minimize an objective function. Two optimization methods will be considered in this chapter: The Levenberg-Marquardt Algorithm (LMA) and the Conjugate Gradient (CG) one. Both algorithms are described in details in this chapter.

3.3 Levenberg-Marquardt Algorithm (LMA)

LMA is an iterative method for solving nonlinear least squares problems which was proposed by Levenberg in 1944 [87] and modified later by Marquardt in 1963 [88]. The LMA proceeds iteratively as follows

$$\mathbf{f}^{(\mathrm{k}+1)} = \mathbf{f}^{(\mathrm{k})} - [(\mathbf{Ja}^{(\mathrm{k})})^{\mathrm{T}}(\mathbf{Ja}^{(\mathrm{k})}) + \mu^{(\mathrm{k})}\mathbf{\Omega}^{(\mathrm{k})}]^{-1}(\mathbf{Ja}^{(\mathrm{k})})^{\mathrm{T}}[T_{i,N} - T_{d}] \tag{3.22}$$

where $\mathbf{f} \equiv \mathbf{x}$ or $\mathbf{y}$ are the x- or y- coordinates of the nodes at the boundary of the physical domain , T denotes transpose of matrix, $\mathbf{Ja}$ is the Jacobian or Sensitivity Matrix, $\mu^{(\mathrm{k})}$ is the damping parameter (positive scalar), and $\mathbf{\Omega}^{(\mathrm{k})}$ is a diagonal matrix in form of

$$\mathbf{\Omega}^{(\mathrm{k})} = \mathrm{diag}[(\mathbf{Ja}^{(\mathrm{k})})^{\mathrm{T}}(\mathbf{Ja}^{(\mathrm{k})})] \tag{3.23}$$

Problems for which $\left|\mathbf{Ja}^{\mathrm{T}}\mathbf{Ja}\right| \approx 0$ are referred to as *ill-conditioned*. In solving systems of equations of the form $\mathbf{AX} = \mathbf{B}$, the coefficient matrix $\mathbf{A}$ is said to be ill-conditioned if any slight changes in the coefficient $\mathbf{B}$ results in a large changes in the solution $\mathbf{X}$. In general, IHTPs are very ill-conditioned particularly near the initial guess used for the unknown parameters which may be far from the actual solution. To circumvent this problem, the term $\mu^{(\mathrm{k})}\mathbf{\Omega}^{(\mathrm{k})}$ is introduced in the update to damp oscillations and instabilities in

the solution of the problem [25, 89]. In another version of the Levenberg-Marquardt method, the matrix $\mathbf{\Omega}^{(\mathrm{k})}$ is taken as the identity matrix [90], namely

$$\mathbf{\Omega}^{(\mathrm{k})} = \mathbf{I} \tag{3.24}$$

In this thesis, both Equations (3.23) and (3.24) were used to evaluate $\mathbf{\Omega}^{(\mathrm{k})}$. The results revealed that the latter version, namely Equation (3.24), is much more efficient. Indeed, the use of this version for LMA resulted in the optimal shape for the problem under consideration. In most cases we considered, the use of Equation (3.23) gave rise to oscillation of the solution and the termination of the optimization process thereafter.

For $\mu^{(\mathrm{k})} \to 0$, the LMA reduces to the *Gauss-Newton method* with its high convergence rate whereas for $\mu^{(\mathrm{k})} \to \infty$, it behaves like the *steepest descent method* with its lower convergence rate but more robust characteristics. Initially $\mu^{(0)}$ is chosen large and as the iterative procedure advances, its value gradually reduces which gives rise to the tendency of the LMA to behave like the Gauss-Newton method [25]. The sensitivity of the objective function $\mathcal{J}$ to changes in the locations of the nodes at the boundary of the domain plays an important role in optimization including use of the LMA. The sensitivity coefficients or arrays of the Jacobian matrix should not be too small as this leads to the problem becoming ill-conditioned. Indeed, small values of the sensitivity coefficients mean that large variations of the locations of the boundary nodes result in insignificant variations of $T_{i,N}$ which yields $\left|\mathbf{Ja}^{\mathrm{T}}\mathbf{Ja}\right| \approx 0$. Hence, this situation can be avoided if the values of sensitivity coefficients are large and uncorrelated [40]. Consequently, by substituting these relationships into the LMA (Equation (3.22)) and following the flowchart given below the optimal shape for the problem under consideration can be obtained.

1. Specify the physical domain, the boundary conditions, the desired outer surface temperature, and the initial value for damping parameter $\mu^{(\mathrm{k})}$ (say, 0.0001 for $\mathrm{k} = 0$).
2. Generate the boundary fitted grid using the grid generation methods described in Chapter 2.

3. Solve the direct problem of finding the temperature values at any grid point of the physical domain.
4. Using Equation (3.5), compute the objective function ($\boldsymbol{\mathcal{J}}^{(k)}$).
5. If value of the objective function obtained in step 4 is less than the specified stopping criterion, the optimization is finished. Otherwise, go to step 6.
6. Compute the sensitivity matrix **Ja** from Equations (3.14) and (3.15).
7. Use LMA (Equation (3.22)) to update $\mathbf{f}$, the location of the boundary nodes.
8. Generate the boundary fitted grid with these new boundary nodes describing a new shape (step 2).
9. Solve the direct problem to find new temperature values at any grid points of the physical domain (step 3).
10. Using Equation (3.5), compute the new objective function ($\boldsymbol{\mathcal{J}}^{(k+1)}$).
11. If ($\boldsymbol{\mathcal{J}}^{(k+1)} \geq \boldsymbol{\mathcal{J}}^{(k)}$), replace $\mu^{(k)}$ by $2\mu^{(k)}$ and return to step 7.
12. If ($\boldsymbol{\mathcal{J}}^{(k+1)} < \boldsymbol{\mathcal{J}}^{(k)}$), accept new values of $\mathbf{f}$ and replace $\mu^{(k)}$ by $0.5\mu^{(k)}$ and return to step 8.
13. Continue until stopping criterion is satisfied.

3.4 Stopping criteria

In this study, the following criteria are used to terminate the iteration in LMA and later in CGM [91].

$$\boldsymbol{\mathcal{J}}^{(k+1)} < \varepsilon_1 \tag{3.25}$$

$$\| \mathbf{f}^{(k+1)} - \mathbf{f}^{(k)} \| < \varepsilon_2 \tag{3.26}$$

where ε_1 and ε_2 are predefined values and $\|\cdot\|$ is the Euclidean norm. Equation (3.25) implies that the objective function is sufficiently small and the boundary shape is adequately close to the desired one. The second criterion, Equation (3.26), relates to the fact that the shape modification after

successive iterations has become minimal. For our shape optimization problems, left hand side of Equation (3.26) may be written as follows

$$\sqrt{(x_{1,N}^{(k+1)} - x_{1,N}^{(k)})^2 + (x_{2,N}^{(k+1)} - x_{2,N}^{(k)})^2 + \cdots + (x_{M,N}^{(k+1)} - x_{M,N}^{(k)})^2}$$

and

$$\sqrt{(y_{1,N}^{(k+1)} - y_{1,N}^{(k)})^2 + (y_{2,N}^{(k+1)} - y_{2,N}^{(k)})^2 + \cdots + (y_{M,N}^{(k+1)} - y_{M,N}^{(k)})^2}$$

However, there exist cases for which by varying the redistribution criteria (they will be introduced later), the grid size, or the initial guess (shape), one can improve the trend of the optimization. In addition, the optimization process may have an oscillatory behavior. In other words, it may sometimes diverges for a few iterations and then continue to converge thereafter.

3.5 Redistribution method

During the iterative solution for updating the outer surface, especially using a coarse grid, some oscillations in the position of the boundary nodes were observed by the onset of ill-ordered nodal points which result in abrupt change of outer profile and hence rapid divergence of the iteration solution (see Figure 3-2).

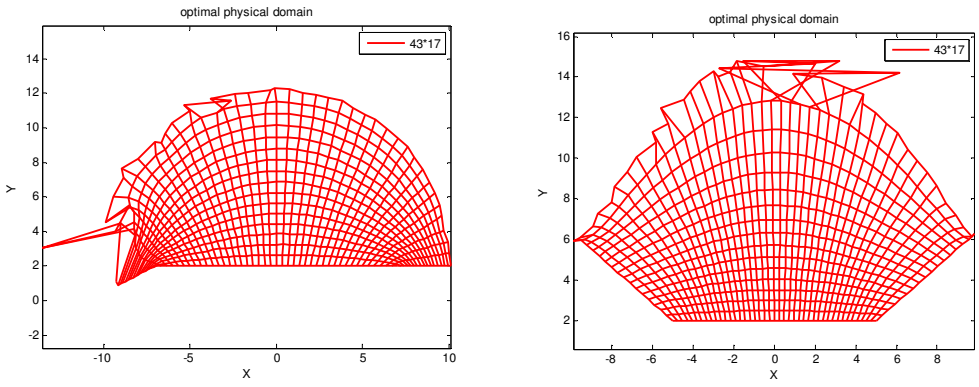


Figure 3-2 ill-ordered nodes at the boundary of the physical domain

To resolve this problem, a novel redistribution procedure was introduced. This method uses criteria to detect ill-ordered nodes, removes

them, and finally replaces them by new well-ordered nodes. The crux of this redistribution method is based on the assumption that grid lines near the "unknown" outer profile are smooth and continuous [92].

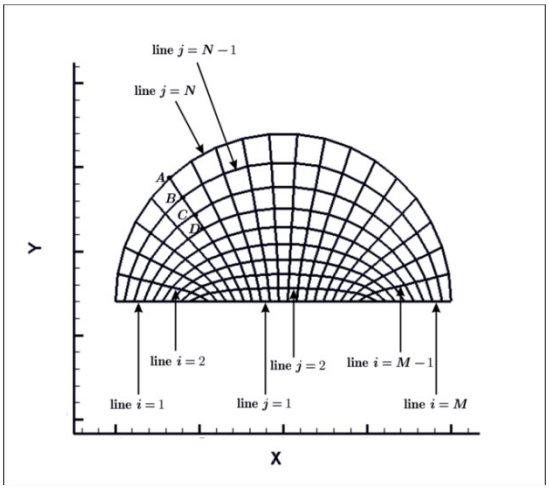


Figure 3-3 scheme illustrating the parameters required to define the redistribution method.

The redistribution method developed here is based on this fact that for any line $i(i = 1,2,3,\ldots,M)$ at least last three segment lines which connect the grid points $(i,N-3)$, $(i,N-2)$, $(i,N-1)$, and (i,N) are collinear and approximately equal. As shown in Figure 3-3, for, say, line $(i=6)$, the segment lines AB, BC, and CD are almost collinear and the ratio $\frac{AB}{BC}$ is approximately equal to the ratio $\frac{BC}{CD}$. Hence, the two criteria which are considered for redistribution of the boundary nodes are (1) the ratio of two successive segment lines on a given line i and (2) the angle between them. For <u>*any*</u> line i, we can compute the ratio of the last three segment lines and the angle between last three segment lines and detect the occurrence of ill-ordered nodes. In this case, all of $x_{i,N}$ and $y_{i,N}$ $(i = 1,2,3,\ldots,M)$ can be removed and the new values of these parameters can be placed based on the two *ratio* and *angle* criteria. For example, suppose that the point A shown in Figure 3-3 is an ill-ordered node. We can remove all the nodes of line $j = N$. Therefore, since ratio $\frac{BC}{CD} = \varepsilon_1$, the place of point A can be determined based on relation $AB = \varepsilon_1 BC$. For any line i, ε_1 has a different value.

Therefore, all nodes on the outer surface (e.g. A) can be tracked based on different values of ε_1. After detecting the node A as an ill-ordered grid point and determine its new place based on $AB = \varepsilon_1 BC$, it is still required to find its orientation. Indeed, the new length of AB is known but its orientation is not yet known. Another criterion for detecting ill-ordered nodes and repositioning them is the angle, as previously pointed out. If the angle between BC and CD is equal to ε_2, then the angle between AB and BC can also be approximated by ε_2. Now with known values of ε_1 and ε_2 for a grid point such as A, it can be easily repositioned. In this study, two criteria are considered to detect ill-ordered nodes. These two criteria are

1. If

$$\frac{L_{(i,N)\to(i,N-1)}}{L_{(i,N-1)\to(i,N-2)}} \geq \lambda_1 \tag{3.27}$$

or

2. If the angle between two successive segment lines

$$L_{(i,N)\to(i,N-1)} \text{ and } L_{(i,N-1)\to(i,N-2)} \geq \lambda_2 \tag{3.28}$$

Here the notation $L_{(i,N)\to(i,N-1)}$ stands for the length of segment line joining two points $\left(i,N\right)$ and $(i,N-1)$.

In most cases, choosing appropriate values for λ_1 and λ_2 is a difficult and time consuming process and is completely dependent on the problem, the boundary conditions, the objective function, and especially the desired outer temperature. From experience, it was noted that for the most of shape optimization problems under consideration, the redistribution method was never used during the optimization process. Indeed, the redistribution parameters λ_1 and λ_2 were implemented in the program only to prevent a probable problem caused by the onset of ill-ordered nodes. Later we will see how the redistribution method detects the ill-ordered nodes, and relocate them to achieve an optimal shape.

3.6 Results for LMA

Before proceeding to the shape optimization using the LMA, it is necessary to fully specify the problem and the associated boundary conditions. Consider the physical domain shown in Figure 3-4.

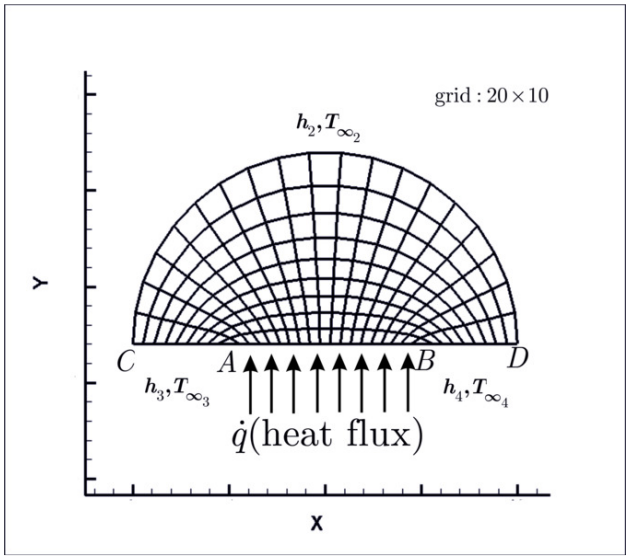


Figure 3-4 The physical domain and the associated boundary conditions used in the shape optimization problem

As shown, the side (boundary) AB is subjected to a constant heat flux, and the other sides are exposed to convective and/or radiative heat transfer. The aim is to find an optimal shape featuring the uniform temperatures on the outer surface (semicircle CD) for the prescribed boundary conditions. The above figure is not restricted to only a semicircle. Indeed, the semicircle illustrated in Figure 3-4 is a representative of any four sided domain. For example, the side involving the heat flux (AB) may be straight or curved (e.g. a semicircle). Furthermore, the sides (AC) and (BD) may be of different size. Several cases are examined and the results are shown as follows.

Figure 3-5 through Figure 3-7 represent the results obtained from the application of LMA in the heat transfer problems specified below. Each figure consists of five plots which represent the initial meshed domain, the optimized meshed domain, the comparison of the initial and optimized domains, the

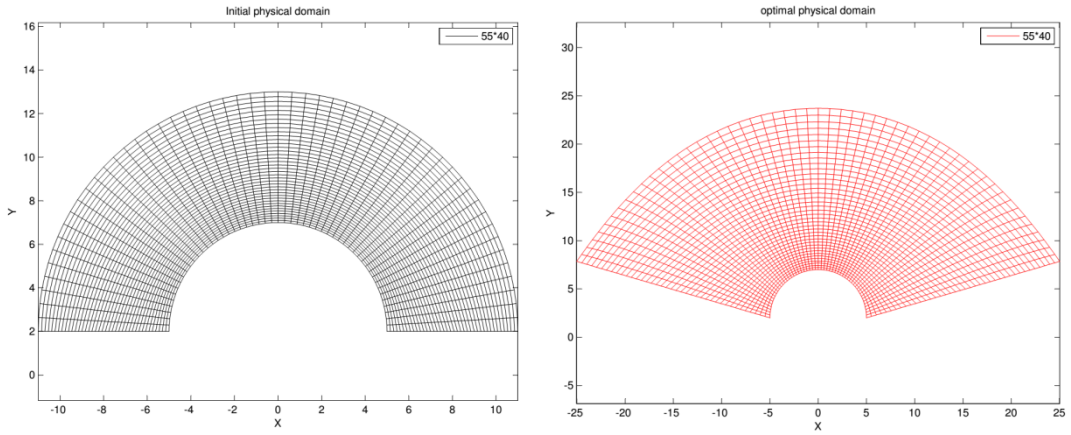
decrease in the objective function versus the iteration number, and the comparison of the temperatures for the outer surface nodes of the initial and optimized domains.

Test case 1 (LMA):

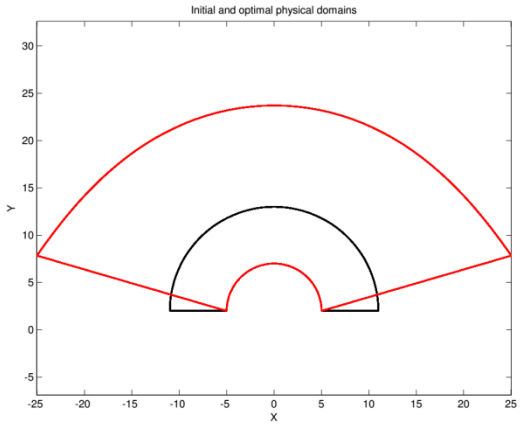
Numerical values of the coefficients involved in Test case 1 are listed in Table 3-1.

$\dot{q}_1 = 500(\frac{\mathrm{W}}{\mathrm{m}^2}), k = 40(\frac{\mathrm{W}}{\mathrm{m.C}}), h_2 = 10(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_3 = 10(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_4 = 10(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}})$
$T_{\infty_2} = 30(^{\circ}\mathrm{C}), T_{\infty_3} = 30(^{\circ}\mathrm{C}), T_{\infty_4} = 30(^{\circ}\mathrm{C}), T_d = 35(^{\circ}\mathrm{C})$

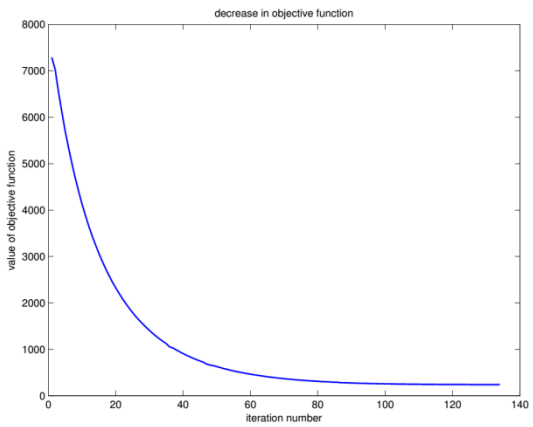
Table 3-1 Data used for Test case 1 (LMA)



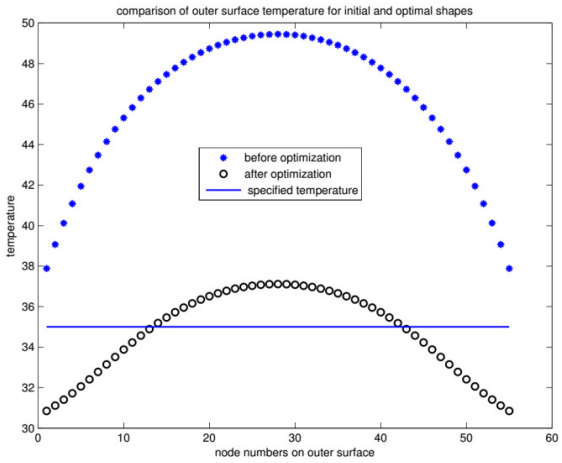
(a) (b)



(c)



(d)



(e)

Figure 3-5 Initial (a) and optimal (b) meshed physical domains, comparison of the initial and optimal meshed physical domains (c), decrease in objective function in terms of the iteration number as a measure for evaluation the convergence rate and the trend of optimization (d), and the comparison of the temperatures for the outer surface nodes for both initial and optimized domains (e).

Figure 3-5a and Figure 3-5b depict the initial and optimal physical domains discretized by a 55×40 mesh using an elliptic grid generation technique. As it can be seen from Figure 3-5c, the variation of the optimal shape to the initial one is significant which results in a considerable and smooth decrease in objective function value during 134 iterations (Figure 3-5d). The decrease is from ~ 7289 to ~ 239 which in turn shows a $\sim \%96.7$ reduction in

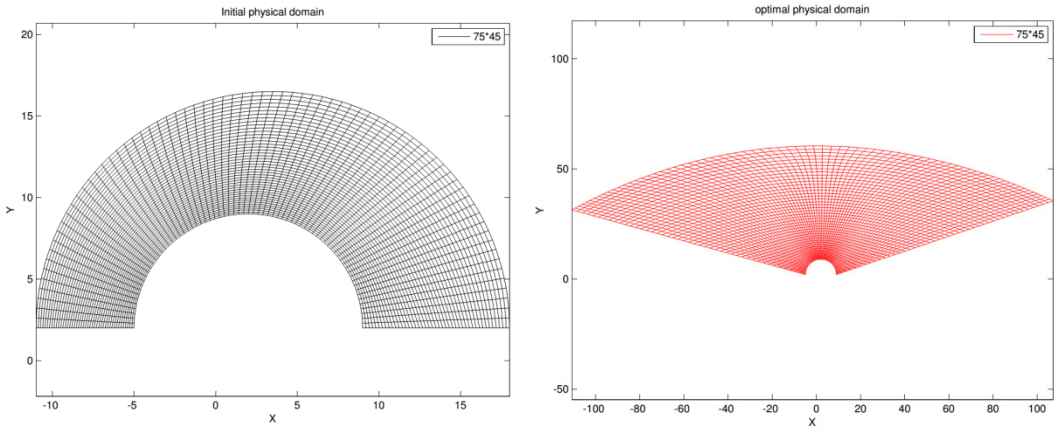
objective function value. The outer surface temperatures values for initial and optimal shapes are demonstrated in Figure 3-5e. This figure shows that the outer surface temperature values for the optimal shape match much more closely the desired temperature. The results are obtained by a Fortran compiler (Force 2.0) and computations are run on a PC with Intel Pentium Dual 1.73 and 1G RAM and the computation time is about 9 min. The tolerance used in the iterative process is 10^{-8} and variables are double precision.

Numerical values of the coefficients involved in Test case 2 and 3 are listed in Table 3-2 and Table 3-4, respectively. The explanations for these test cases are the same as Test case 1. The results for Test case 2 and 3 are depicted in Figure 3-6 and Figure 3-7, respectively.

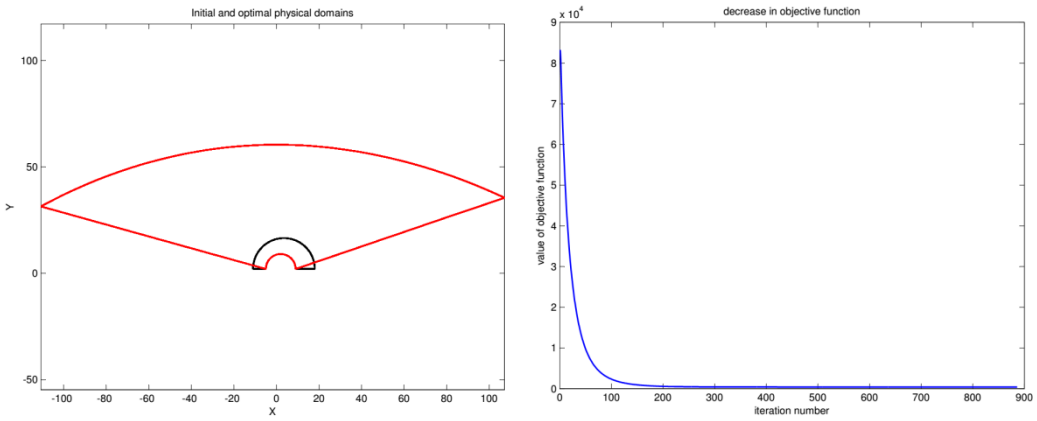
Test case 2 (LMA):

$\dot{q}_1 = 800(\frac{\mathrm{W}}{\mathrm{m}^2}), k = 50(\frac{\mathrm{W}}{\mathrm{m.C}}), h_2 = 8(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_3 = 8(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_4 = 8(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}})$
$T_{\infty_2} = 20(^\circ\mathrm{C}), T_{\infty_3} = 20(^\circ\mathrm{C}), T_{\infty_4} = 20(^\circ\mathrm{C}), T_d = 23(^\circ\mathrm{C})$

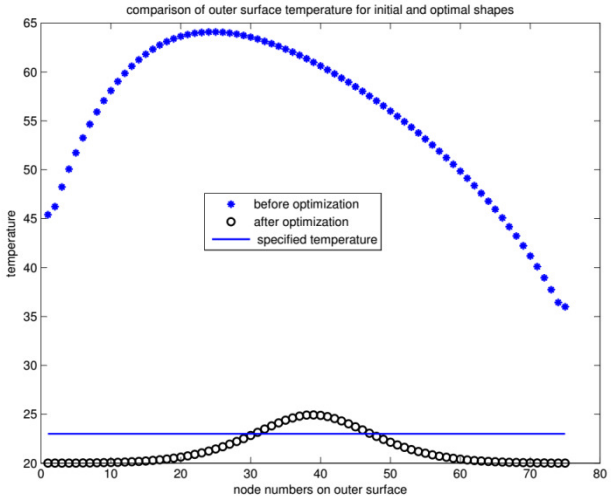
Table 3-2 Data used for Test case 2 (LMA)



(a) (b)



(c) (d)



(e)

Figure 3-6 Initial (a) and optimal (b) meshed physical domains, comparison of the initial and optimal meshed physical domains (c), decrease in objective function in terms of the iteration number as a measure for evaluation the convergence rate and the trend of optimization (d), and the comparison of the temperatures for the outer surface nodes for both initial and optimized domains (e).

The summary of results for Test case 2 is given in Table 3-3.

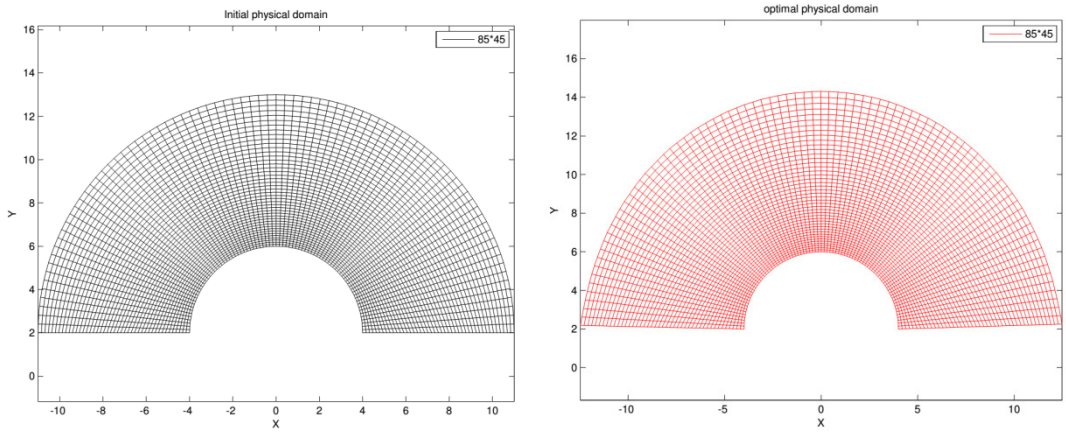
Decrease in objective function	From 83222.25 to 383.60
Percentage of decrease in objective function	%99.5
Computation time	28.5 min
Number of iterations	886

Table 3-3 Summary of results for Test case 2 (LMA)

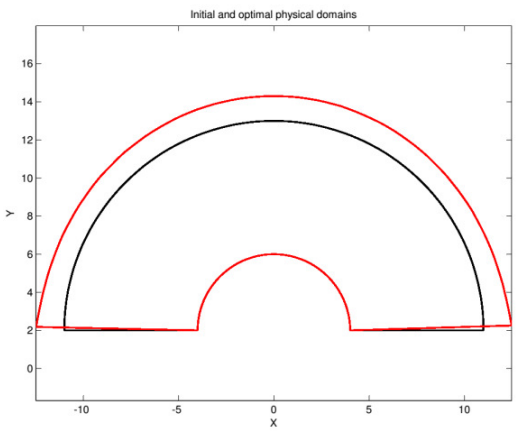
Test case 3 (LMA):

$\dot{q}_1 = 500(\frac{\mathrm{W}}{\mathrm{m}^2}), k = 40(\frac{\mathrm{W}}{\mathrm{m.C}}), h_2 = 3(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_3 = 0(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_4 = 0(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}})$
$T_{\infty_2} = 30(^{\circ}\mathrm{C}), T_{\infty_3} = 30(^{\circ}\mathrm{C}), T_{\infty_4} = 30(^{\circ}\mathrm{C}), T_d = 85(^{\circ}\mathrm{C})$

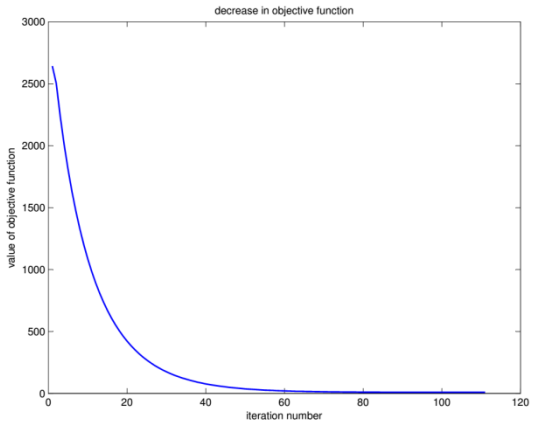
Table 3-4 Data used for Test case 3 (LMA)



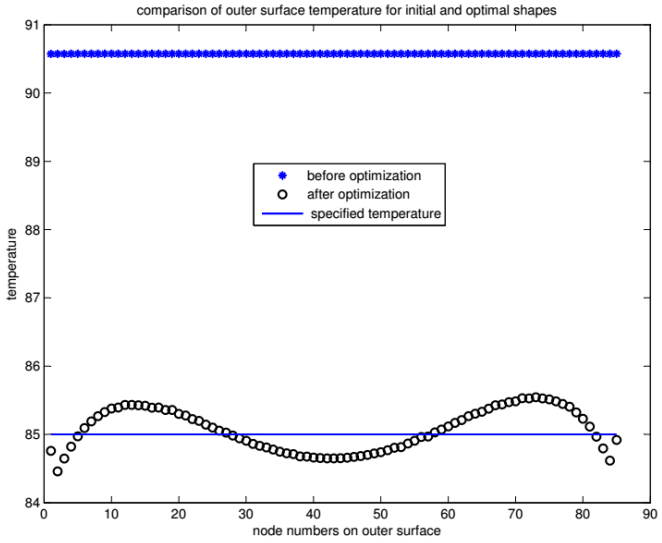
(a) (b)



(c)



(d)



(e)

Figure 3-7 Initial (a) and optimal (b) meshed physical domains, comparison of the initial and optimal meshed physical domains (c), decrease in objective function in terms of the iteration number as a measure for evaluation the convergence rate and the trend of optimization (d), and the comparison of the temperatures for the outer surface nodes for both initial and optimized domains (e).

The summary of results for Test case 3 is given in Table 3-5.

Decrease in objective function	From 2642.49 to 8.20
Percentage of decrease in objective function	%99.7
Computation time	45.5 min
Number of iterations	111

Table 3-5 Summary of results for Test case 3 (LMA)

As can be seen from Table 3-4, the values for h_3 and h_4 are zero. In other words, two sides AC and BD in Figure 3-4 are insulated. In such a case, the heat conduction problem reduces to a one dimensional problem in the radial direction. Therefore, it is expected that the optimal shape design is also a semicircle. Since the desired temperature ($T_d = 85(^{\circ}\mathrm{C})$) is smaller than the constant outer temperature ($\sim 90.5^{\circ}\mathrm{C}$), it is expected that the outer surface for the optimal shape is farther from the heat flux side than the one for the initial shape. In other words, the radius of the optimal shape should be bigger than the one of the initial shape (see Figure 3-7c).

An analytical solution for Test case 3 will be given here to validate the shape optimization algorithm employed. As pointed out previously, Test case 3 is reduced to one dimensional heat conduction problem by insulating both sides adjacent to the side subject to the heat flux. It is, therefore, similar to the problem of steady heat conduction through a circular pipe. Consider Figure 3-8 which shows a one directional heat conduction problem in which the heat is conducted across two semicircles surfaces and then dissipated to the ambient by convection. The heat flux for this problem is [49]

$$\dot{q} = \frac{\dot{Q}}{A} = \frac{\pi L (T_1 - T_{\infty_2}) \Big/ \left[\dfrac{\ln(r_2 / r_1)}{k} + \dfrac{1}{r_2 h_2} \right]}{\pi r_1 L} = \frac{(T_1 - T_{\infty_2})}{r_1 \left[\dfrac{\ln(r_2 / r_1)}{k} + \dfrac{1}{r_2 h_2} \right]}$$

where L is the length of the semicircular pipe, and $\dot{Q}$ is the rate of heat conduction. Solving the above equation for r_2 gives a nonlinear expression. Thus, we substitute the obtained result for r_2 from the optimization process

and check whether the value for heat flux will be $500(\frac{\mathrm{W}}{\mathrm{m}^2})$ (see Table 3-4) or not. From Figure 3-7b, we can see that the value of r_2 varies between 12.30 and 12.47. Therefore, if we assume the average of these two values for r_2, namely $12.385\,\mathrm{m}$, then we will have

$$\dot{q} = \frac{(141-30)}{4\left[\frac{\ln(\frac{12.385}{4})}{40} + \frac{1}{(12.385)(3)}\right]} = 503(\frac{\mathrm{W}}{\mathrm{m}^2})$$

which is in an excellent agreement with known value of $500(\frac{\mathrm{W}}{\mathrm{m}^2})$. Hence the optimal radius (r_2) is about $12.385\,\mathrm{m}$ thereby confirming the optimization process accuracy using LMA.

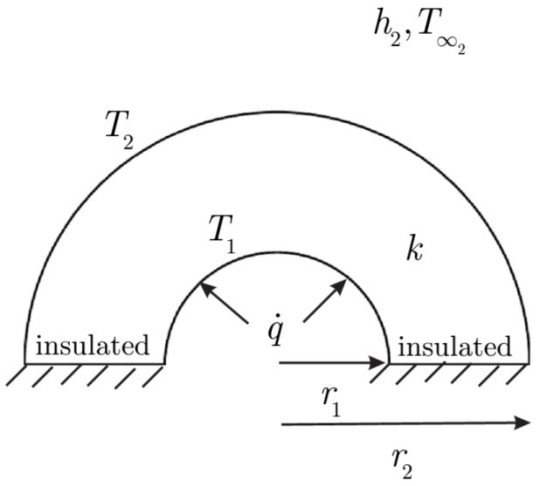


Figure 3-8 One dimensional (in radial direction) heat conduction across two semicircles surfaces.

3.7 Conjugate Gradient Method (CGM)

Conjugate Gradient Method (CGM) is another powerful optimization method employed in this thesis. No matrix inversion or storage of a large $n \times n$

matrix is required for the CGM. The CGM can be successfully applied to linear and nonlinear optimization problems. It is robust, reliable, and has a high rate of convergence. It requires a small amount of computation [89, 91, 93-98]. Therefore, it is an ideal gradient based optimization method for the shape optimization of interest here. The mathematical formulation of the CGM can be found in the above references and here we only mention the iterative algorithm. In this method, the objective function given by Equation (3.5) is minimized by searching along *the direction of descent* $\mathbf{d}^{(\mathrm{k})}$ using a *search step size* $\beta^{(\mathrm{k})}$.

$$\mathbf{f}^{(\mathrm{k}+1)} = \mathbf{f}^{(\mathrm{k})} - \beta^{(\mathrm{k})}\mathbf{d}^{(\mathrm{k})} \tag{3.29}$$

where $\mathbf{f}$ is the vector containing the unknown coordinates of the boundary nodes to be optimized .The direction of descent of the current iteration is obtained as a linear combination of the direction of descent of the previous iteration and the gradient direction $\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})}$. In other words, it is a conjugation of the gradient direction of the current iteration, $\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})}$, and the direction of descent of the previous iteration, $\mathbf{d}^{(\mathrm{k}-1)}$, accordingly

$$\mathbf{d}^{(\mathrm{k})} = \nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})} + \gamma^{(\mathrm{k})}\mathbf{d}^{(\mathrm{k}-1)} \tag{3.30}$$

where $\gamma^{(\mathrm{k})}$ is the conjugation coefficient. There exist several formulae for the calculation of the conjugation coefficient [25, 94, 97, 99-101] :

1. The Fletcher-Reeves formula [99]:

$$\gamma^{(\mathrm{k})} = \frac{\left[\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})}\right]^{\mathrm{T}} \nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})}}{\left[\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}-1)}\right]^{\mathrm{T}} \nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}-1)}} \tag{3.31}$$

2. The Polak-Ribiere formula [100]:

$$\gamma^{(\mathrm{k})} = \frac{\left[\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})}\right]^{\mathrm{T}} (\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})} - \nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}-1)})}{\| \nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}-1)} \|^2} = \frac{\left[\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})}\right]^{\mathrm{T}} (\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})} - \nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}-1)})}{\left[\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}-1)}\right]^{\mathrm{T}} \nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}-1)}} \tag{3.32}$$

The gradient direction $\nabla \boldsymbol{\mathcal{J}}^{(\mathrm{k})}$ is obtained by differentiating the objective function with respect to the unknown coordinates of the boundary nodes which was given by Equations (3.6) and (3.7). By minimizing the objective function $\boldsymbol{\mathcal{J}}$ with respect to the search step size $\beta^{(\mathrm{k})}$, one can find an expression for calculating the search step size as follows [25]

$$\beta^{(\mathrm{k})} = \frac{[\mathbf{Ja}^{(\mathrm{k})}\mathbf{d}^{(\mathrm{k})}]^{\mathrm{T}}[T_{i,N} - T_d]}{[\mathbf{Ja}^{(\mathrm{k})}\mathbf{d}^{(\mathrm{k})}]^{\mathrm{T}}[\mathbf{Ja}^{(\mathrm{k})}\mathbf{d}^{(\mathrm{k})}]} \tag{3.33}$$

By the algorithm given below one can obtain the optimal shape for the problem under consideration.

1. Specify the physical domain, the boundary conditions, and the desired outer surface temperature.
2. Generate the boundary fitted grids using the grid generation methods described in Chapter 2.
3. Solve the direct problem of finding the temperature values at any grid points of the physical domain.
4. Using Equation (3.5), compute the objective function ($\boldsymbol{\mathcal{J}}^{(\mathrm{k})}$).
5. If value of the objective function obtained in step 4 is less than the specified stopping criterion, the optimization is finished. Otherwise, go to step 6.
6. Compute the sensitivity matrix $\mathbf{Ja}$ from Equations (3.14) and (3.15).
7. Compute the gradient direction $\nabla \boldsymbol{\mathcal{J}}^{(\mathrm{k})}$ from Equations (3.6) and (3.7).
8. Compute the conjugation coefficient $\gamma^{(\mathrm{k})}$ either from Equation (3.31) or Equation (3.32). For $\mathrm{k} = 0$, set $\gamma^{(0)} = 0$.
9. Compute the direction of descent $\mathbf{d}^{(\mathrm{k})}$ from Equation (3.30).
10. Compute the search step size $\beta^{(\mathrm{k})}$ from Equation (3.33).
11. From Equation (3.29), evaluate the new coordinates of boundary nodes $\mathbf{f}^{(\mathrm{k}+1)}$.
12. Set the next iteration ($\mathrm{k} = \mathrm{k} + 1$) and return to the step 2.

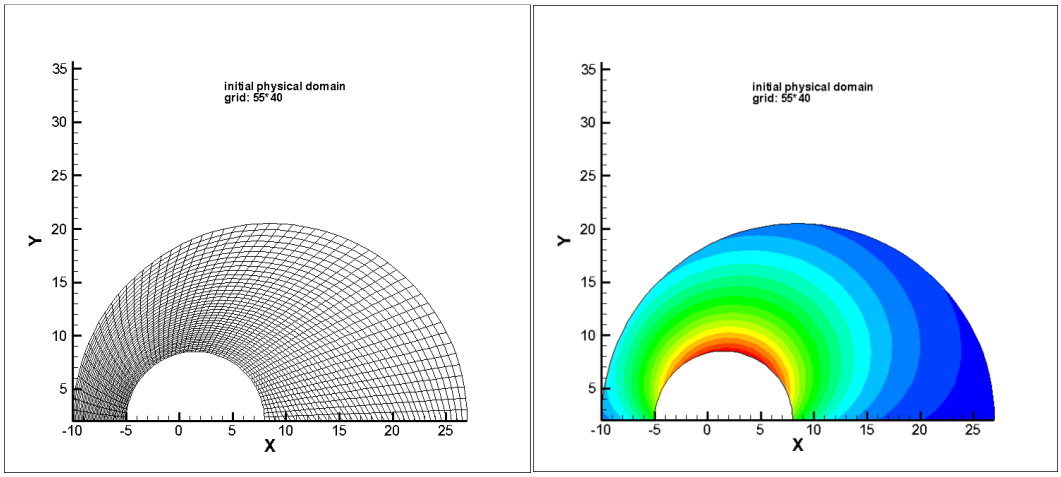
3.8 The results for CGM

The following figures represent the results obtained from the application of CGM in heat conduction problems specified below. Each figure consists of the plots which represent the initial meshed domain, the optimized meshed domain, the combination of the initial and optimized domains, the decrease in the objective function versus the iteration number, and the comparison of the temperatures for the outer surface nodes of the initial and optimized domains.

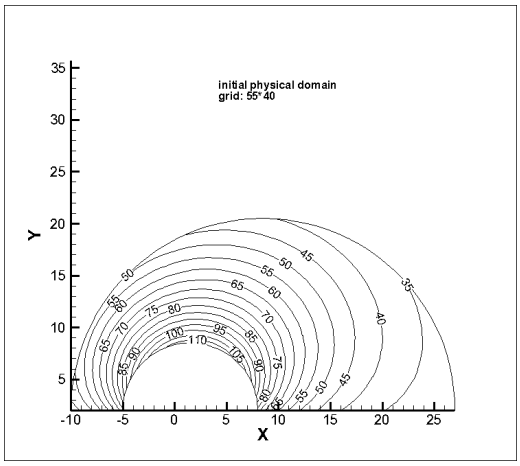
Test case 1 (CGM):

$\dot{q}_1 = 500(\frac{\mathrm{W}}{\mathrm{m}^2}), k = 40(\frac{\mathrm{W}}{\mathrm{m.C}}), h_2 = 10(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_3 = 10(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_4 = 10(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}})$
$T_{\infty_2} = 30(^{\circ}\mathrm{C}), T_{\infty_3} = 30(^{\circ}\mathrm{C}), T_{\infty_4} = 30(^{\circ}\mathrm{C}), T_d = 35(^{\circ}\mathrm{C})$

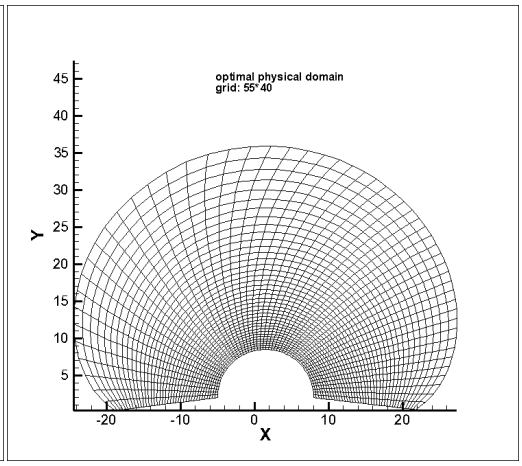
Table 3-6 Data used for Test case 1 (CGM)



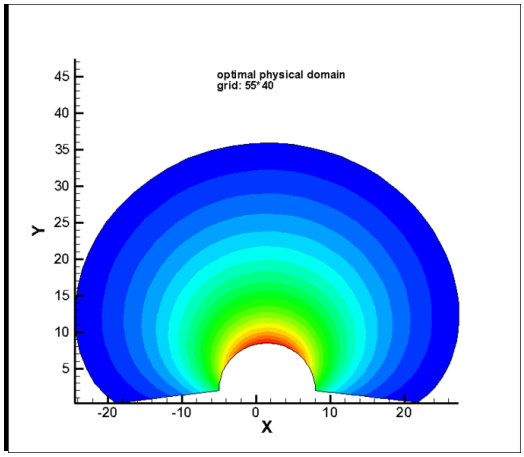
a) b)


initial physical domain
grid: 55* 40
X
Y

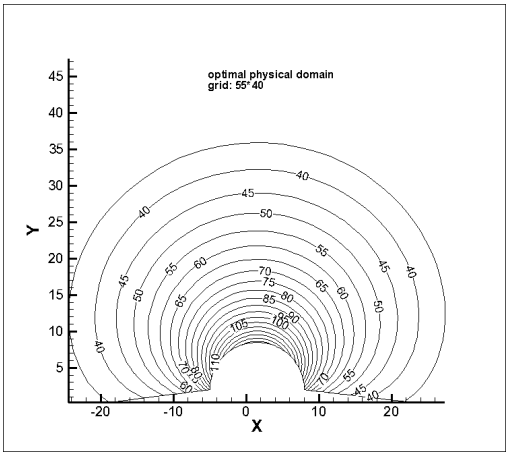
c)


optimal physical domain
grid: 55* 40
X
Y

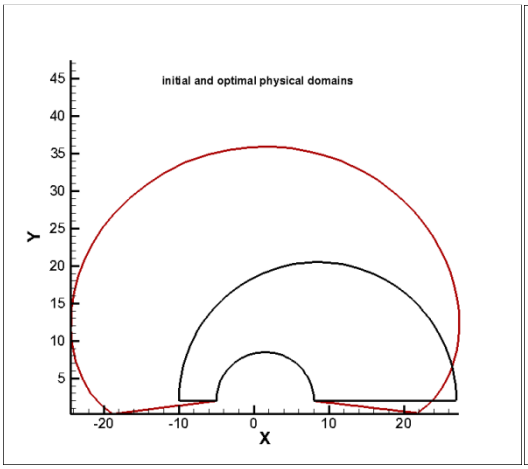
d)


optimal physical domain
grid: 55* 40
X
Y

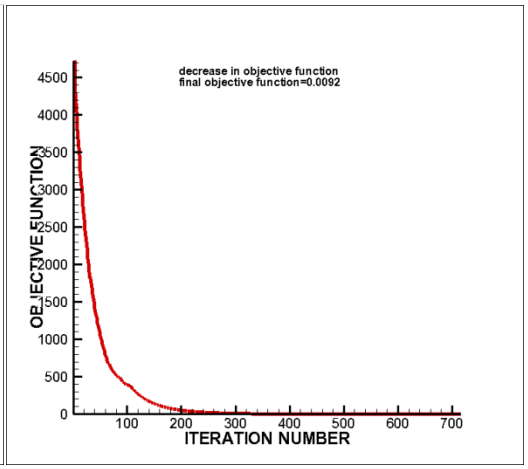
e)


optimal physical domain
grid: 55* 40
X
Y

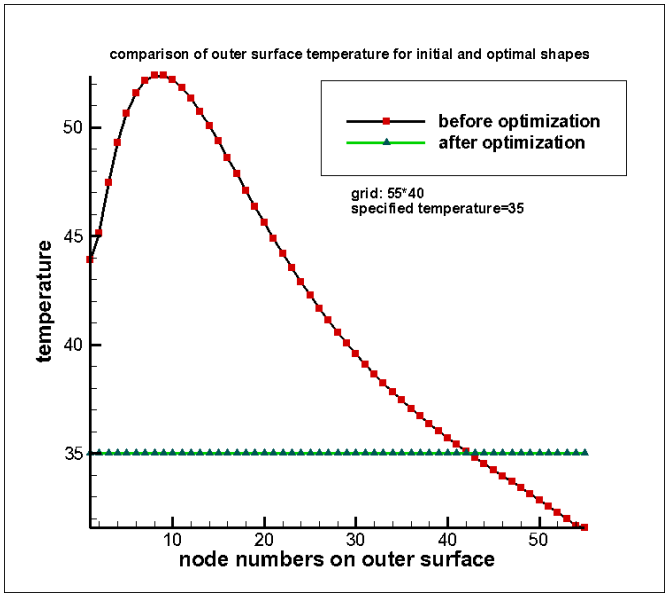
f)


initial and optimal physical domains
X
Y

g)


decrease in objective function
final objective function=0.0092
OBJECTIVE FUNCTION
ITERATION NUMBER

h)



i)

Figure 3-9 Results for Test case 1. a) initial physical domain mesh; b) initial physical domain temperature contours; c) numerical temperature contours for the initial physical domain; d) optimal physical domain mesh; e) optimal physical domain temperature contours; f) numerical temperature contours for the optimal physical domain; g) comparison of the initial and optimal physical domains; h) decrease in objective function; i) comparison of the outer surface temperatures for the initial and optimal shapes.

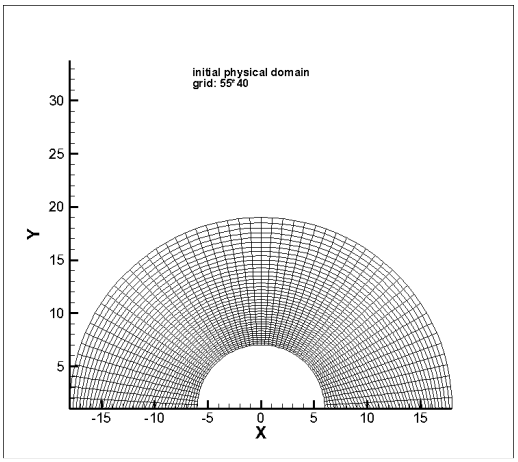
Figure 3-9a depicts the initial physical domain discretized by a 55×40 mesh using elliptic grid generation technique. The temperature contours of the domain are shown in Figure 3-9b and c. It can be clearly seen that the outer surface temperature distribution is not uniform and does not match the desired one ($35^{\circ}\mathrm{C}$). The contour shown in Figure 3-9c shows that the temperature values at the left side of the outer surface (large semicircle) are bigger than the desired temperature and the temperature values at the right side are approximately the same as the desired one. Therefore, more modification is expected in the left side of the initial physical domain, as shown in Figure 3-9g. Furthermore, since the desired temperature is less than the outer surface temperature, especially in the left side, the development of the outer boundary (large semicircle) is away from the boundary involving the heat flux (small semicircle). In other words, the radius of the large semicircle increases. Figure 3-9d and e demonstrate the meshed optimal domain and the temperature contours, respectively. The numerical

temperature contours of the optimal domain are shown in Figure 3-9f. The uniformity of the temperature contours shown in Figure 3-9e and f and exact matching the last contour (outer surface temperature) to the desired temperature is the proof of accuracy of the implemented algorithm. The decrease in objective function versus iteration number is shown in Figure 3-9h. As shown, the objective function decreases from 4730.1 to 0.0092 . This also validates the implementation of the proposed algorithm. The comparison of the outer surface temperatures before and after optimization is depicted in Figure 3-9i. Before optimization, the outer surface temperatures vary from 52.41313 to 31.58256 whereas they range from 34.99766 to 35.02288 after optimization which shows the accuracy of the algorithm. The results are obtained by a Fortran compiler (Force 2.0) and computations are run on a PC with Intel Pentium Dual 1.73 and 1G RAM and the computation time is about 20min. The rest of the plots are devoted to sensitivity coefficients. The results shown in Figure 3-9 confirm the efficiency and accuracy of the employed shape optimization method. To investigate further the utilized method, the following three test cases are investigated. Since only the physical domain and boundary conditions are varied, the explanation for these test cases are the same as Test case 1 (CGM).

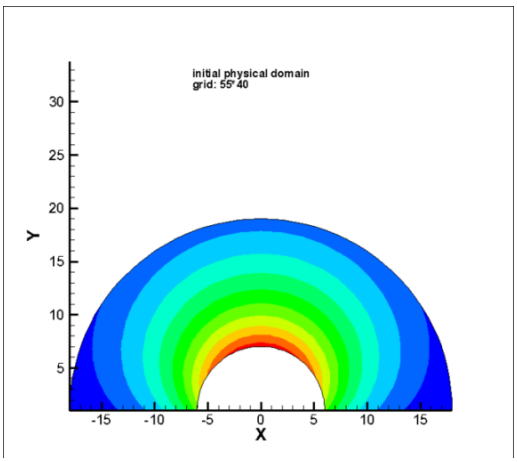
Test case 2 (CGM):

$\dot{q}_1 = 800(\frac{\mathrm{W}}{\mathrm{m}^2}), k = 50(\frac{\mathrm{W}}{\mathrm{m.C}}), h_2 = 8(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_3 = 8(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_4 = 8(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}})$
$T_{\infty_2} = 20(^\circ\mathrm{C}), T_{\infty_3} = 20(^\circ\mathrm{C}), T_{\infty_4} = 20(^\circ\mathrm{C}), T_d = 35(^\circ\mathrm{C})$

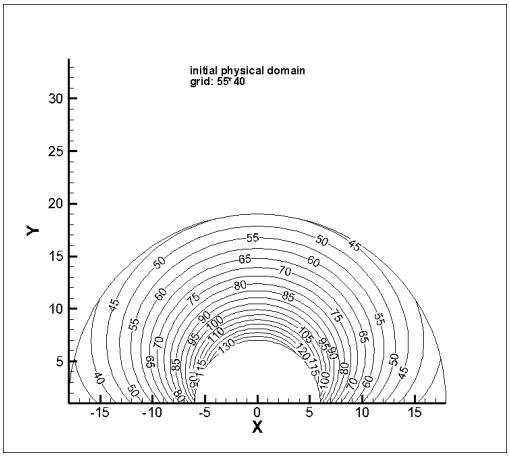
Table 3-7 Data used for Test case 2 (CGM)


initial physical domain
grid: 55* 40
Y
X

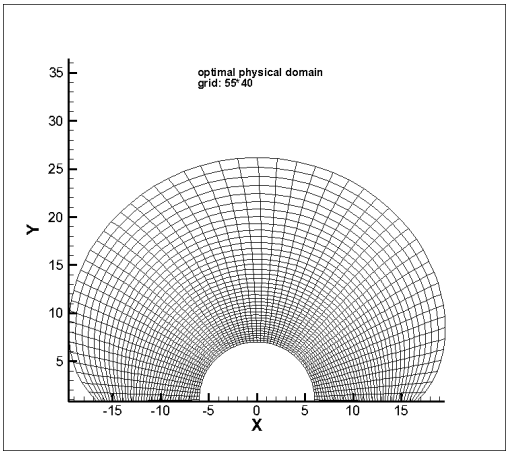
a)


initial physical domain
grid: 55* 40
Y
X

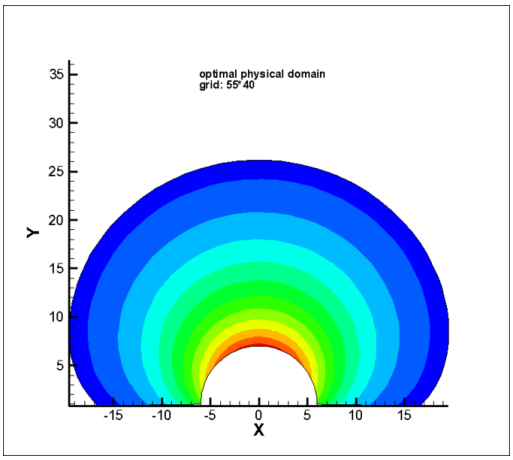
b)


initial physical domain
grid: 55* 40
Y
X

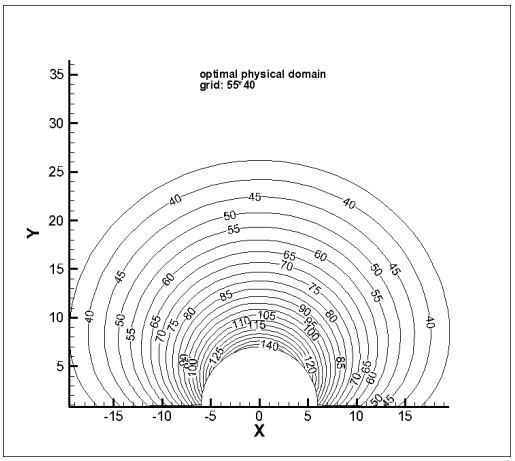
c)


optimal physical domain
grid: 55* 40
Y
X

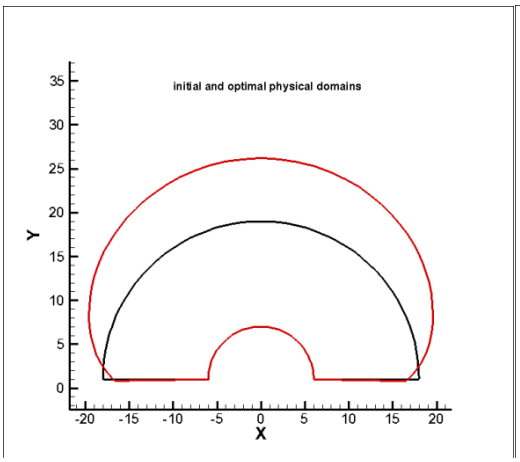
d)


optimal physical domain
grid: 55* 40
Y
X

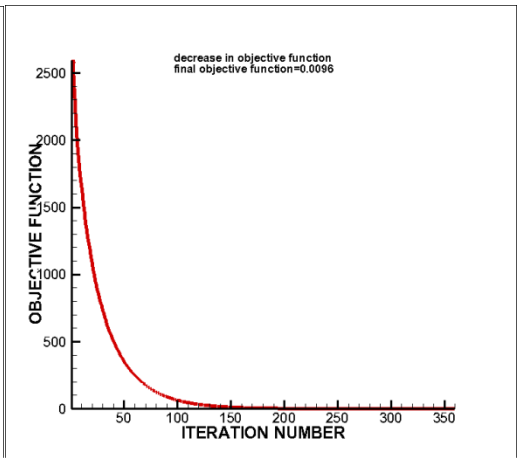
e)


optimal physical domain
grid: 55* 40
Y
X

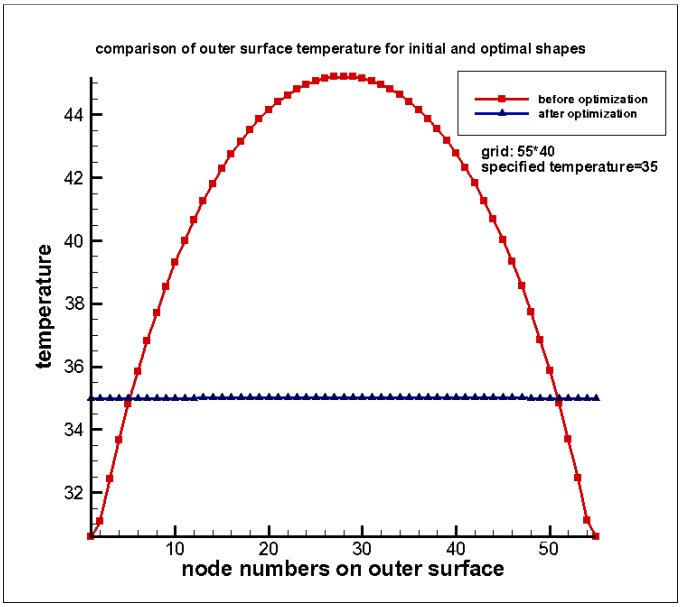
f)



g)



h)



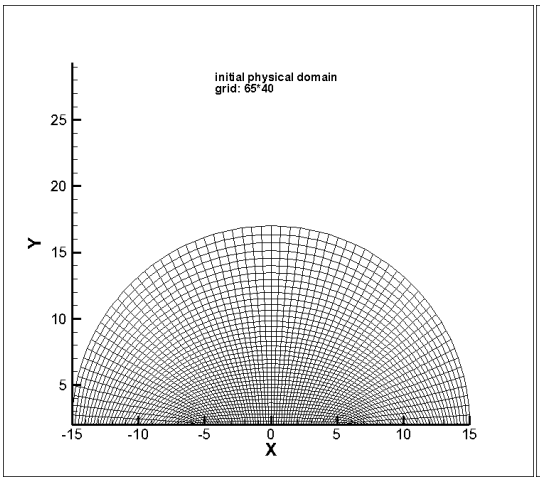
i)

Figure 3-10 Results for Test case 2. a) initial physical domain mesh; b) initial physical domain temperature contours; c) numerical temperature contours for the initial physical domain; d) optimal physical domain mesh; e) optimal physical domain temperature contours; f) numerical temperature contours for the optimal physical domain; g) comparison of the initial and optimal physical domains; h) decrease in objective function; i) comparison of the outer surface temperatures for the initial and optimal shapes.

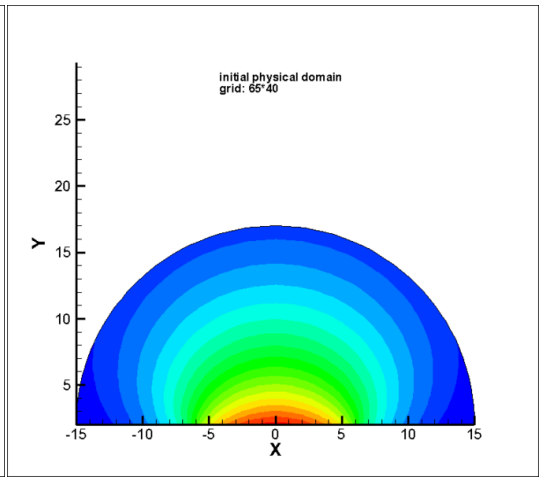
Test case 3 (CGM):

$\dot{q}_1 = 900(\frac{\mathrm{W}}{\mathrm{m}^2}), k = 70(\frac{\mathrm{W}}{\mathrm{m.C}}), h_2 = 6(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_3 = 6(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_4 = 6(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}})$
$T_{\infty_2} = 25(^{\circ}\mathrm{C}), T_{\infty_3} = 25(^{\circ}\mathrm{C}), T_{\infty_4} = 25(^{\circ}\mathrm{C}), T_d = 45(^{\circ}\mathrm{C})$

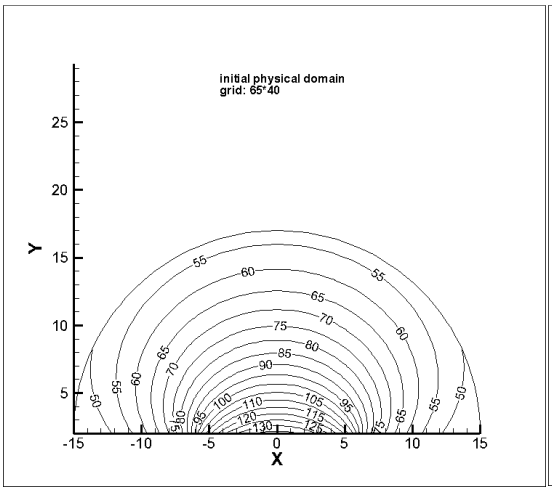
Table 3-8 Data used for Test case 3 (CGM)



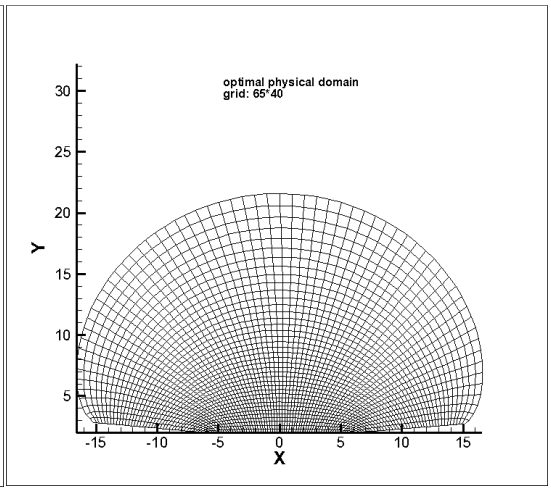
a)



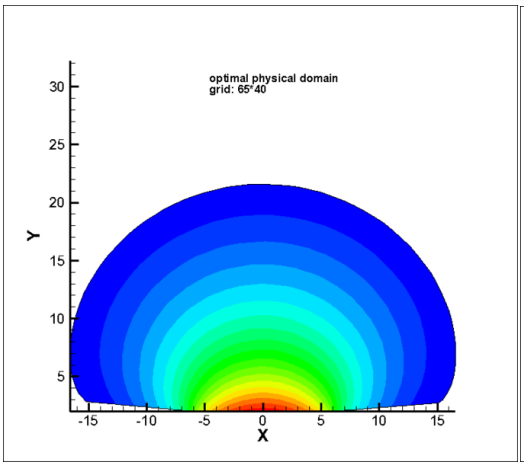
b)



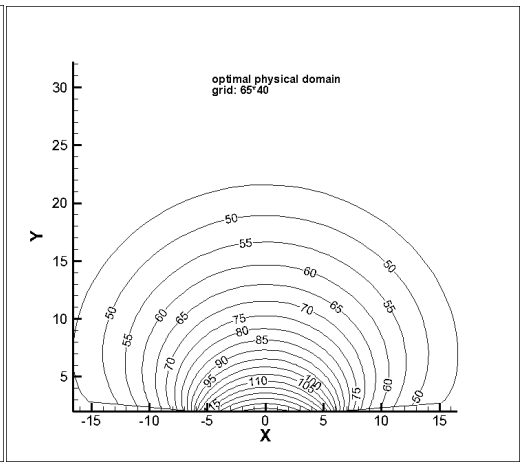
c)



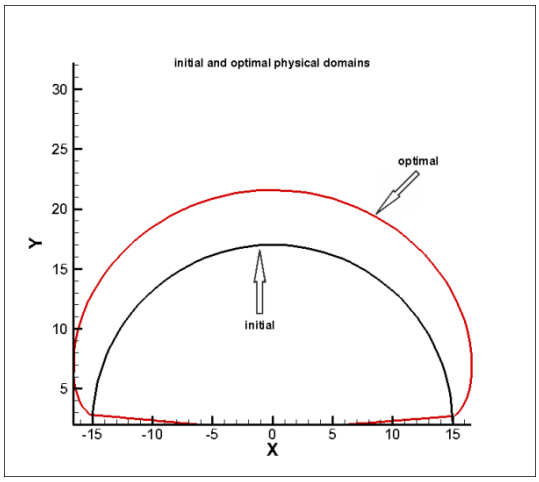
d)


optimal physical domain
grid: 65*40
Y
X

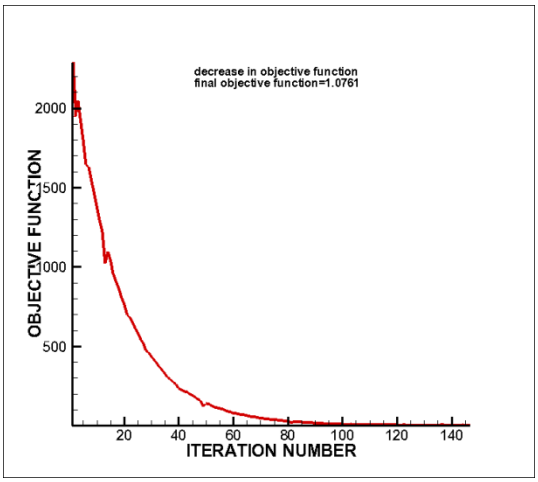
e)


optimal physical domain
grid: 65*40
Y
X

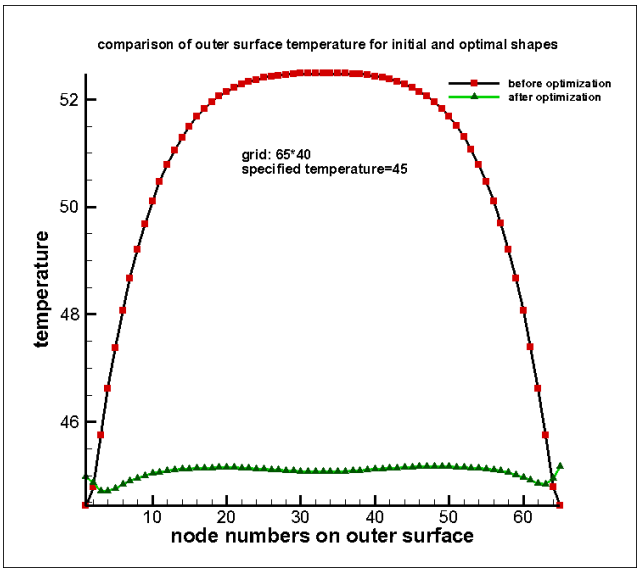
f)


initial and optimal physical domains
optimal
initial
Y
X

g)


decrease in objective function
final objective function=1.0761
OBJECTIVE FUNCTION
ITERATION NUMBER

h)



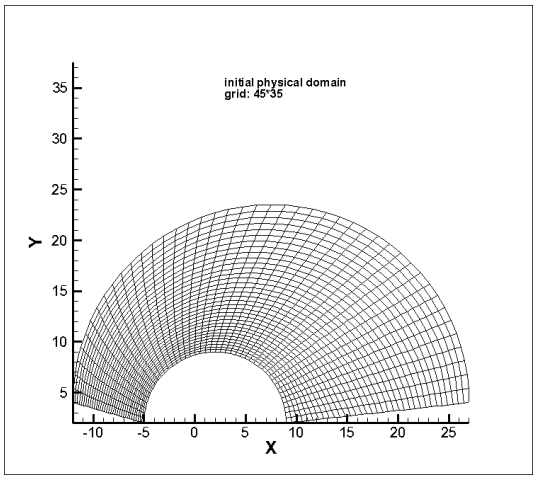
i)

Figure 3-11 Results for Test case 3. a) initial physical domain mesh; b) initial physical domain temperature contours; c) numerical temperature contours for the initial physical domain; d) optimal physical domain mesh; e) optimal physical domain temperature contours; f) numerical temperature contours for the optimal physical domain; g) comparison of the initial and optimal physical domains; h) decrease in objective function; i) comparison of the outer surface temperatures for the initial and optimal shapes.

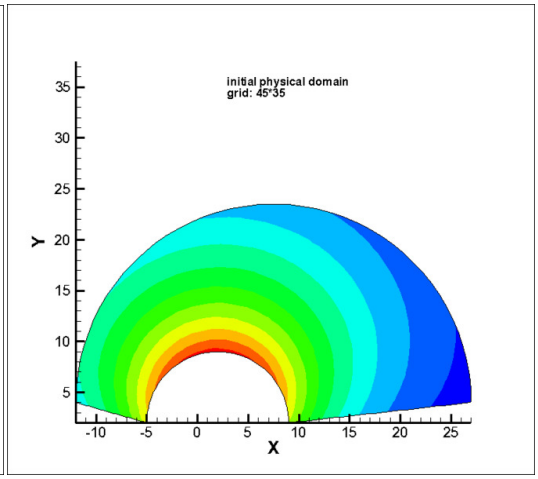
Test case 4 (CGM):

$\dot{q}_1 = 900(\frac{\mathrm{W}}{\mathrm{m}^2}), k = 70(\frac{\mathrm{W}}{\mathrm{m.C}}), h_2 = 6(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_3 = 6(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_4 = 6(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}})$
$T_{\infty_2} = 40(^{\circ}\mathrm{C}), T_{\infty_3} = 40(^{\circ}\mathrm{C}), T_{\infty_4} = 40(^{\circ}\mathrm{C}), T_d = 55(^{\circ}\mathrm{C})$

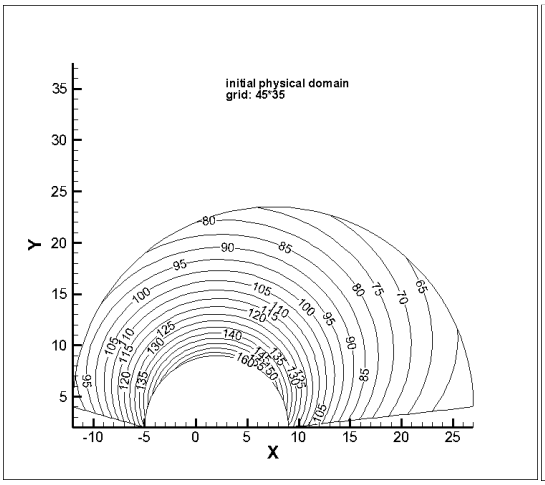
Table 3-9 Data used for Test case 4 (CGM)


initial physical domain
grid: 45*35
Y
X

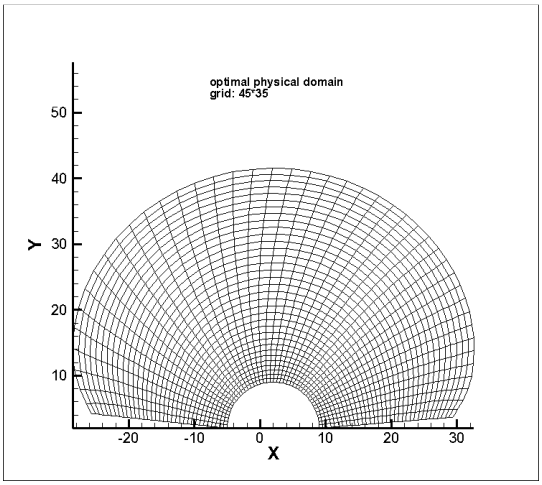
a)


initial physical domain
grid: 45*35
Y
X

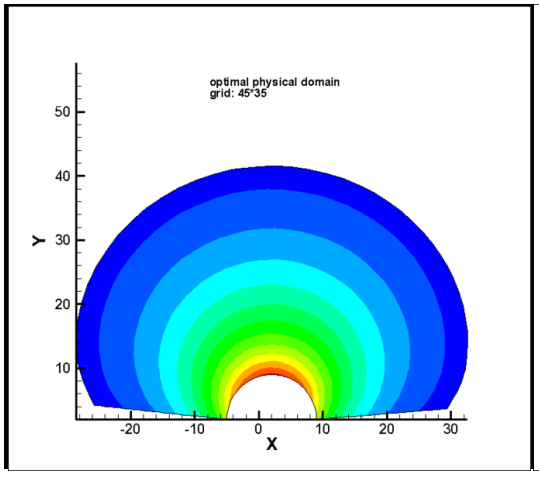
b)


initial physical domain
grid: 45*35
Y
X

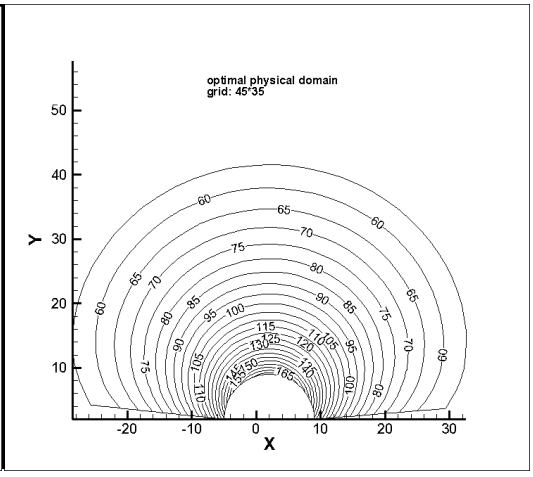
c)


optimal physical domain
grid: 45*35
Y
X

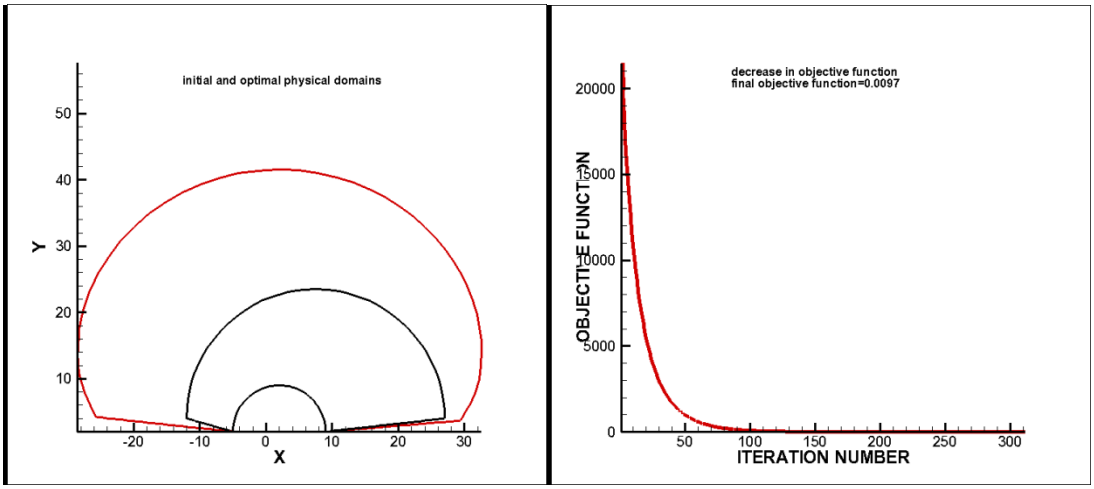
d)


optimal physical domain
grid: 45*35
Y
X

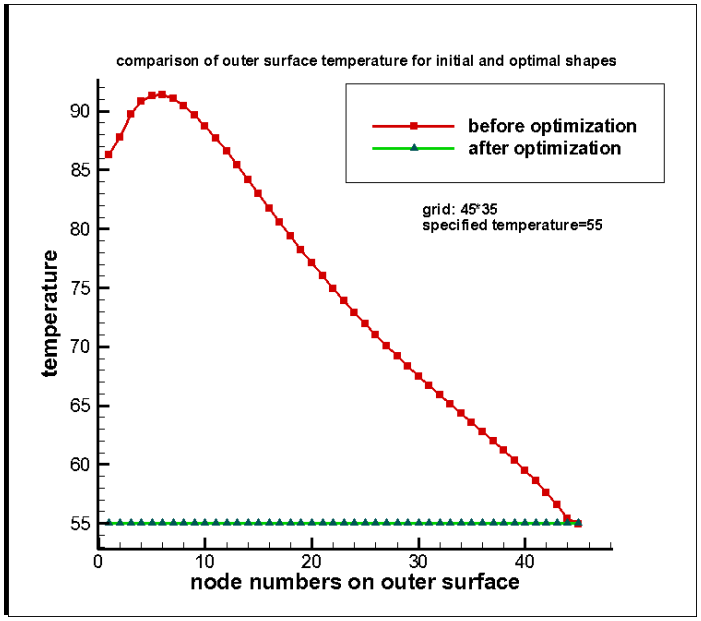
e)


optimal physical domain
grid: 45*35
Y
X

f)



g) h)



i)

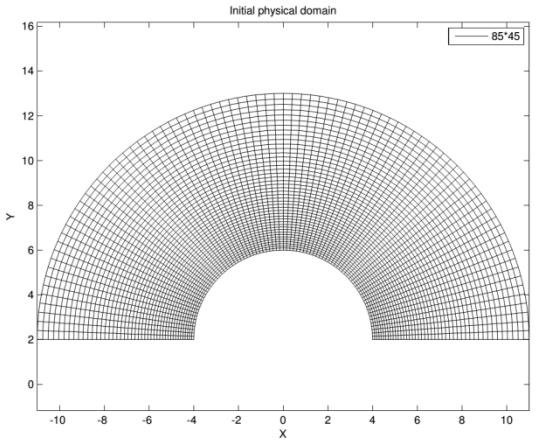
Figure 3-12 Results for Test case 4. a) initial physical domain mesh; b) initial physical domain temperature contours; c) numerical temperature contours for the initial physical domain; d) optimal physical domain mesh; e) optimal physical domain temperature contours; f) numerical temperature contours for the optimal physical domain; g) comparison of the initial and optimal physical domains; h) decrease in objective function; i) comparison of the outer surface temperatures for the initial and optimal shapes.

Test case 5 (CGM):

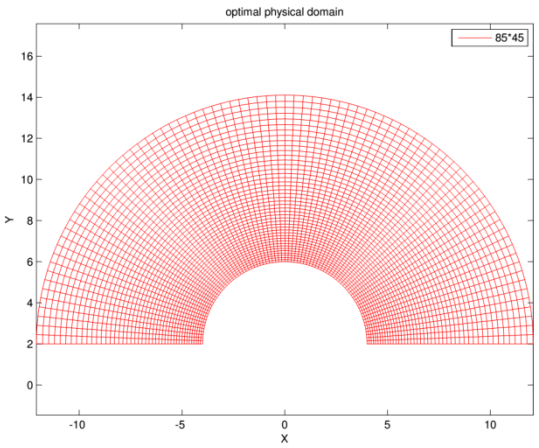
The objective of Test case 5 is twofold. 1) To validate the optimal shape design algorithm using CGM and 2) to compare the performances of the CGM and LMA. Therefore, the data in Test case 3 (LMA) are reemployed here.

$\dot{q}_1 = 500(\frac{\mathrm{W}}{\mathrm{m}^2}), k = 40(\frac{\mathrm{W}}{\mathrm{m.C}}), h_2 = 3(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_3 = 0(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_4 = 0(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}})$
$T_{\infty_2} = 30(^{\circ}\mathrm{C}), T_{\infty_3} = 30(^{\circ}\mathrm{C}), T_{\infty_4} = 30(^{\circ}\mathrm{C}), T_d = 85(^{\circ}\mathrm{C})$

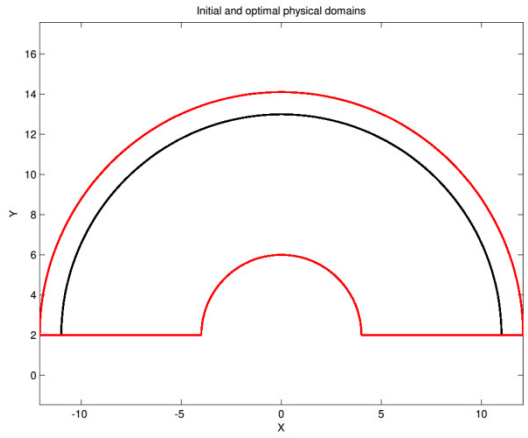
Table 3-10 Data used for Test case 5 (CGM)



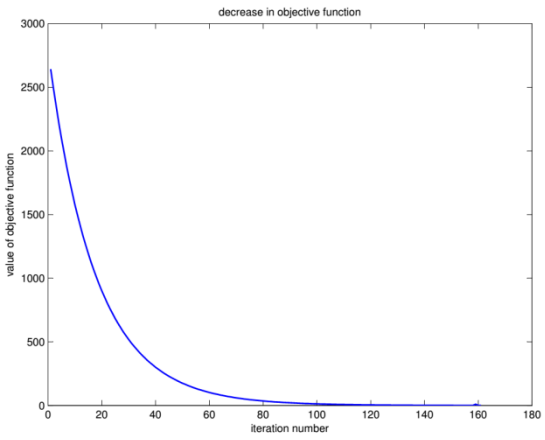
a)



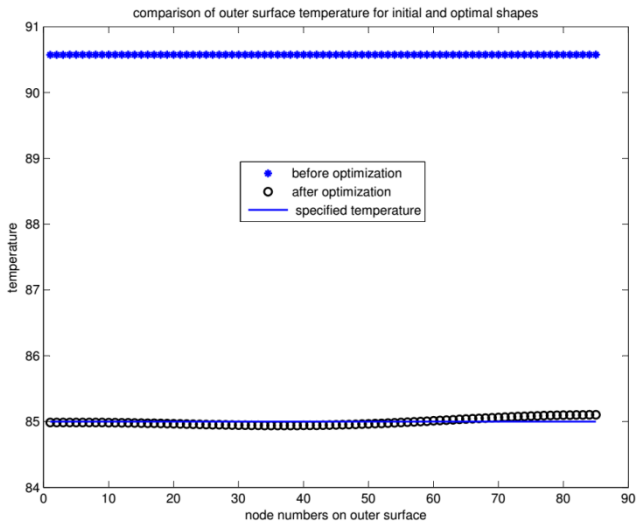
b)



c)



d)



e)

Figure 3-13 Initial (a) and optimal (b) meshed physical domains, comparison of the initial and optimal meshed physical domains (c), decrease in objective function in terms of the iteration number as a measure for evaluation the convergence rate and the trend of optimization (d), and the comparison of the temperatures for the outer surface nodes for both initial and optimized domains (e).

Table 3-10 represent the data used for Test case 5 (CGM) which are the same as in Test case 3 (LMA). The results shown in Figure 3-13 confirm correctness and accuracy of the proposed algorithm. It can be also seen that the CGM works better than LMA in inferring the optimal shape. The CGM reduces the objective function more than LMA. Following the same analytical procedure as in Test case 3 (LMA), a value of $500.5(\frac{\mathrm{W}}{\mathrm{m}^2})$ is obtained for the heat flux. Therefore, it is in excellent agreement with the value of $500(\frac{\mathrm{W}}{\mathrm{m}^2})$. A summary of the results in Test case 5 (CGM) as well as the comparison of performance of CGM and LMA for two identical problems (Test case 3 (LMA) and Test case 5 (CGM)) are given in the following tables. Furthermore, a comparison of the rate of convergence for both methods is given in Figure 3-14. It shows that the LMA has a higher rate of convergence. However, it could not proceed further than 111 iterations.

Decrease in objective function	From 2642.49 to 0.23
Percentage of decrease in objective function	%100
Computation time	1h & 10min
Number of iterations	160

Table 3-11 Summary of results for Test case 5 (CGM).

	CGM	LMA
Decrease in objective function	From 2642.49 to 0.23	From 2642.49 to 8.20
Percentage of decrease in objective function	%100	%99.7
Computation time	1h & 10min	45.5min
Number of iterations	160	111

Table 3-12 Comparison of performance of the CGM and LMA.

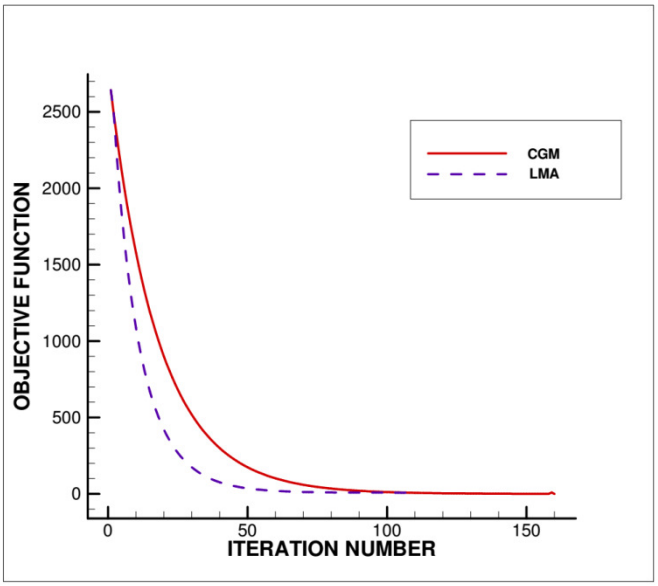


Figure 3-14 Comparison of the rate of convergence for CGM and LMA.

Test case 6 (CGM): The effect of the redistribution scheme

This test case shows the results for a physical domain with five sides (Figure 3-17a). The data used for this test case is given in Table 3-13.

$\dot{q}_1 = 700(\frac{\mathrm{W}}{\mathrm{m}^2}), k = 35(\frac{\mathrm{W}}{\mathrm{m.C}}), h_2 = 7(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_3 = 7(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_4 = 7(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}})$
$T_{\infty_2} = 30(^\circ\mathrm{C}), T_{\infty_3} = 30(^\circ\mathrm{C}), T_{\infty_4} = 30(^\circ\mathrm{C}), T_d = 43(^\circ\mathrm{C})$

Table 3-13 Data used for Test case 6 (CGM)

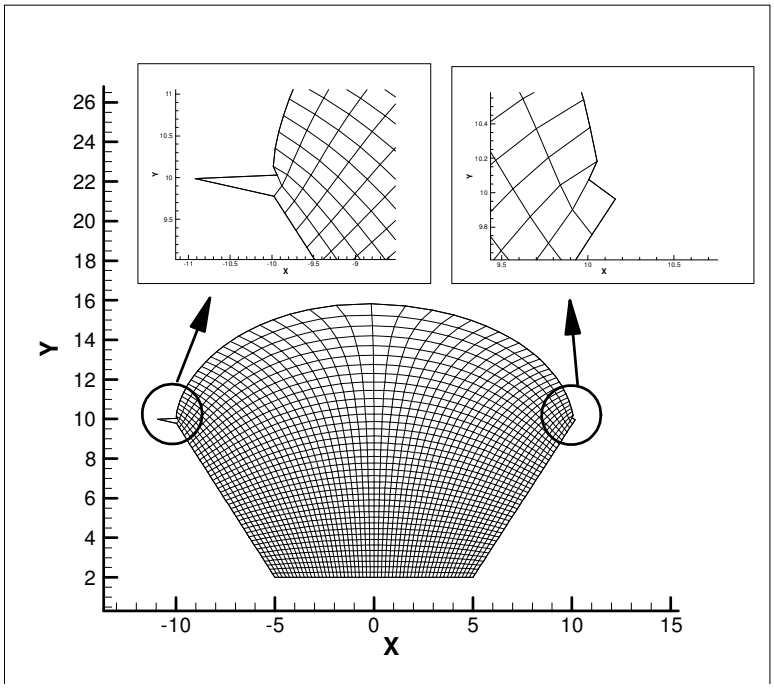


Figure 3-15 Ill-ordered nodes on the outer surface which cause termination of the optimization process.

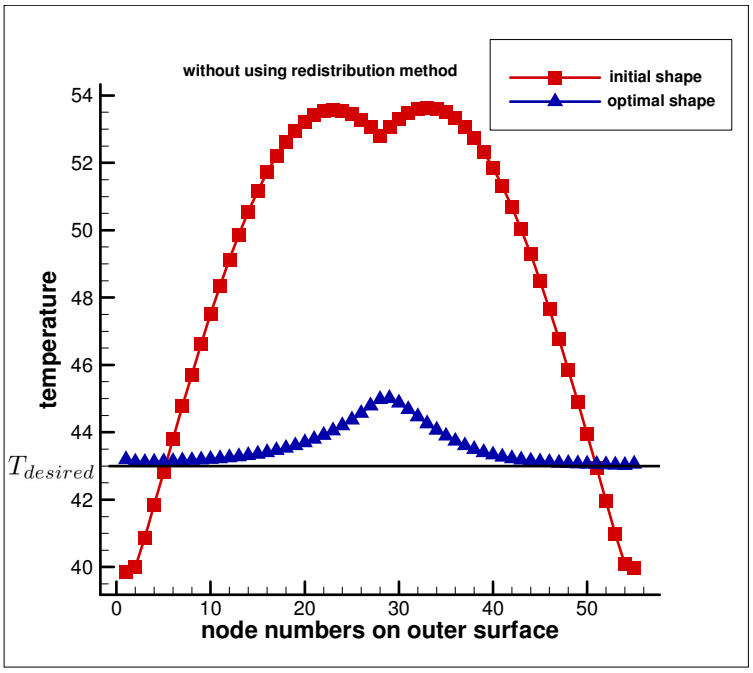
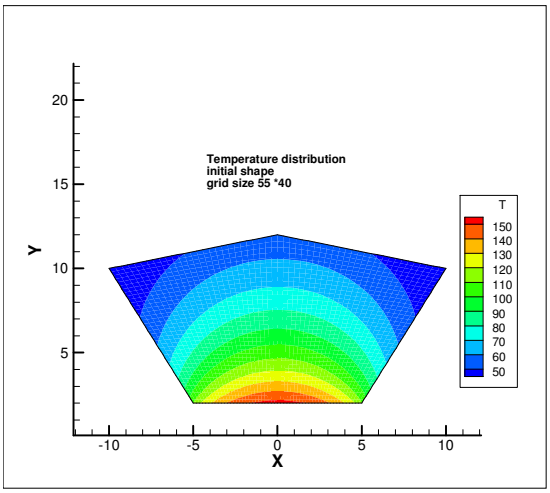
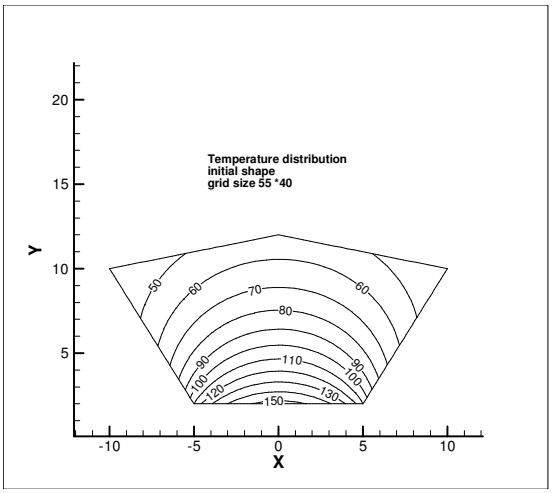


Figure 3-16 Comparison of the temperatures for the outer surface nodes for both initial and optimal shapes without using the redistribution scheme.

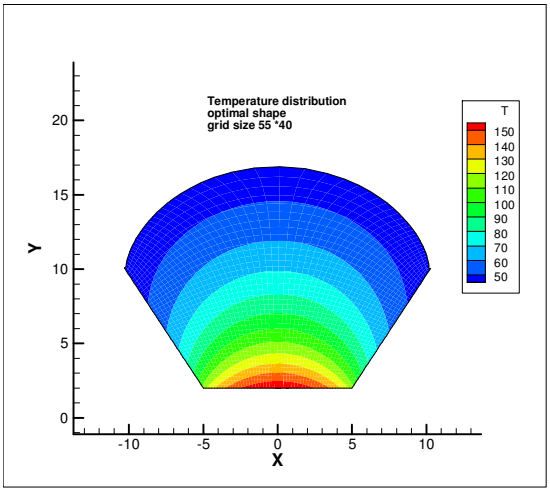
Without using the redistribution scheme, the optimization process advances to the iteration number 101 only where the objective function value is 35.32 (corresponding to %99 reduction in objective function value). The optimization process is stopped at this iteration number due to the onset of oscillation. Figure 3-15 shows the ill-ordered nodes on the outer surface. Thus, the redistribution criteria $\lambda_1 = 4.0$ and $\lambda_2 = 30$ are used to alleviate the oscillation and reach the optimal shape. Introducing the redistribution method causes the optimization process to advance to iteration number 220 where the objective function value is 0.22 (corresponding to %100 reduction in objective function value). The redistribution scheme is employed twice for this test case at iterations 102 and 104. The results for this test case (using the redistribution method) shown in Figure 3-17 reveal how the redistribution scheme can be implemented in the solution procedure to detect and fix the probable oscillation in the location of the outer surface nodes even though it is not used in the optimization process. In other words, it only becomes effective when the oscillation is initiated. The summary of the results is given in Table 3-14.


Temperature distribution
initial shape
grid size 55 *40
T
150
140
130
120
110
100
90
80
70
60
50
Y
X

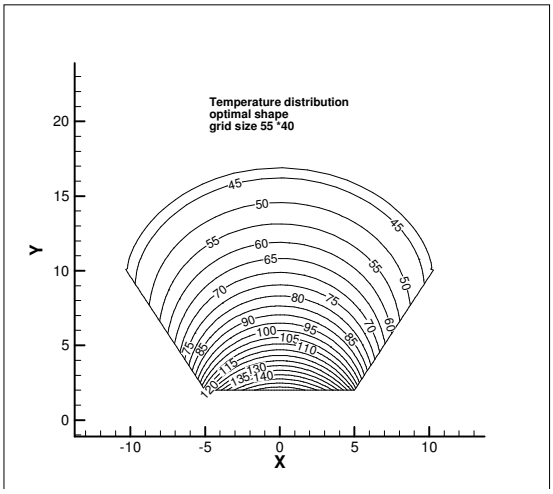
a)


Temperature distribution
initial shape
grid size 55 *40
Y
X

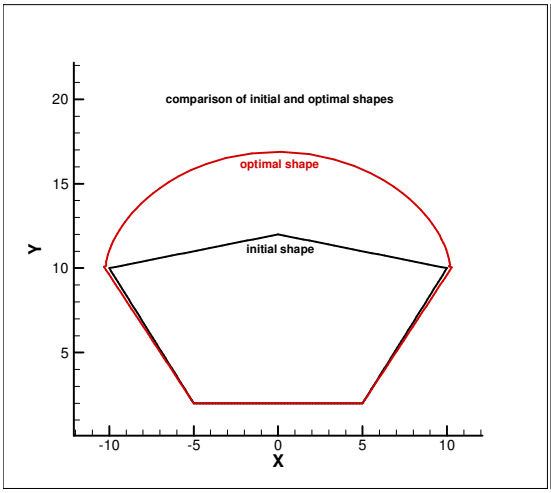
b)


Temperature distribution
optimal shape
grid size 55 *40
T
150
140
130
120
110
100
90
80
70
60
50
Y
X

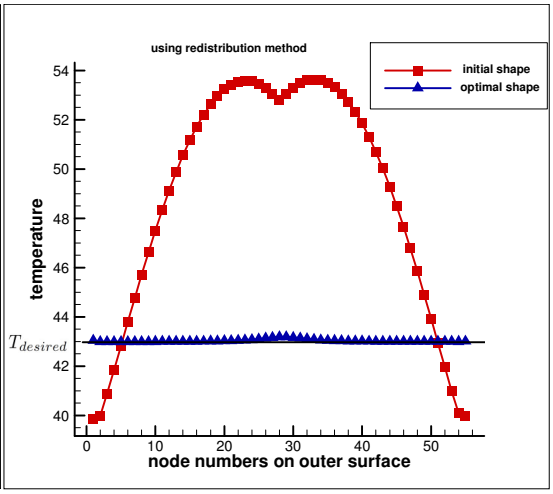
c)


Temperature distribution
optimal shape
grid size 55 *40
Y
X

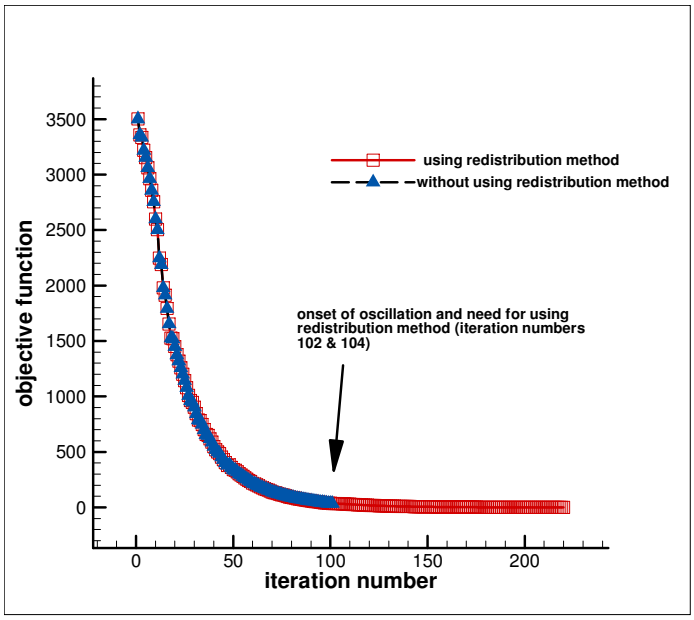
d)


comparison of initial and optimal shapes
optimal shape
initial shape
Y
X

e)


using redistribution method
initial shape
optimal shape
temperature
$T_{desired}$
node numbers on outer surface

f)



g)

Figure 3-17 Temperature distribution for the initial shape [(a),(b)] and the optimal shape [(c),(d)], comparison of the initial and optimal shapes (e), the comparison of the temperatures for the outer surface nodes for both initial and optimal shapes (f), and objective function versus iteration number (g).

Decrease in objective function	From 3501.40 to 0.22
Percentage of decrease in objective function	%100
Computation time	4 min : 7 seconds
Number of iterations	220

Table 3-14 Summary of results for Test case 6 (CGM)

Test case 7 (CGM): The variable heat flux

This test case is concerned with a variable heat flux which increases linearly from 0 at node $i=1$ to $\dot{q}$ at node $i=M$ (Figure 3-18). Data for Test case 7 is given in Table 3-15. The results are shown in Figure 3-19.

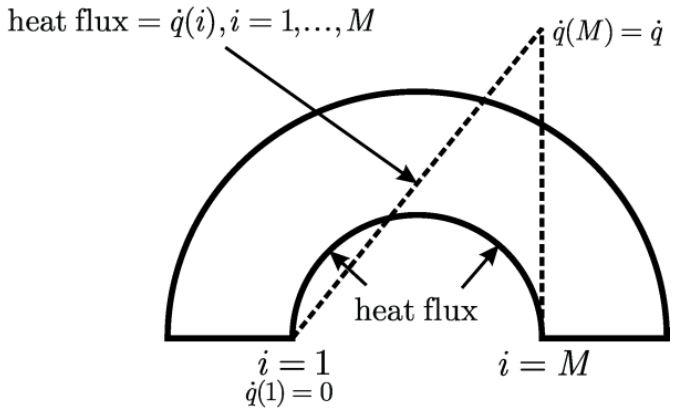
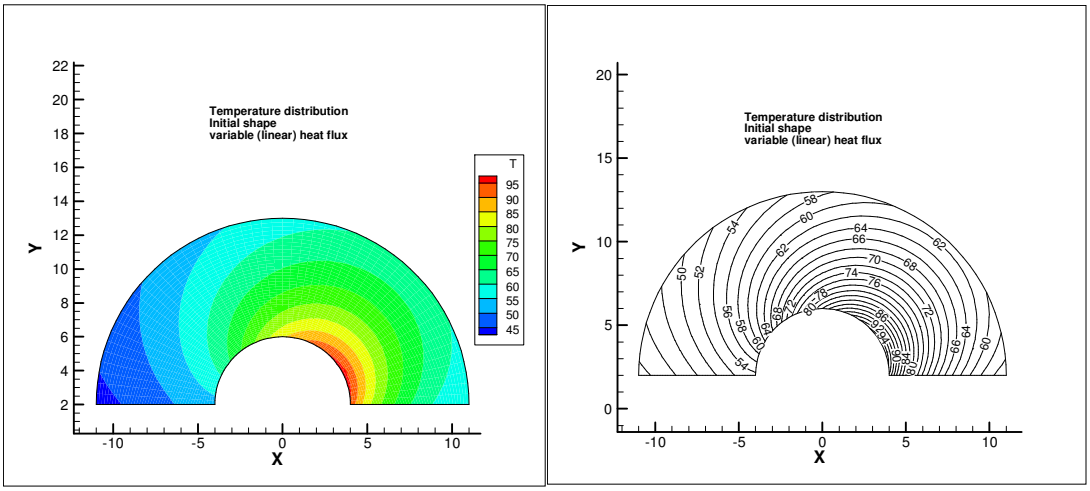


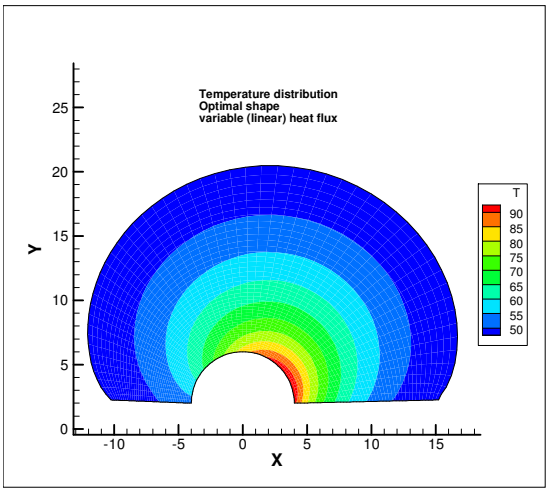
Figure 3-18 Variable heat flux. The heat flux increases linearly from 0 at node $i = 1$ to $\dot{q}$ at node $i = M$.

$\dot{q}_1(M) = 600(\frac{\mathrm{W}}{\mathrm{m}^2}), k = 40(\frac{\mathrm{W}}{\mathrm{m.C}}), h_2 = 3(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_3 = 3(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_4 = 3(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}})$
$T_{\infty_2} = 30(^{\circ}\mathrm{C}), T_{\infty_3} = 30(^{\circ}\mathrm{C}), T_{\infty_4} = 30(^{\circ}\mathrm{C}), T_d = 45(^{\circ}\mathrm{C})$

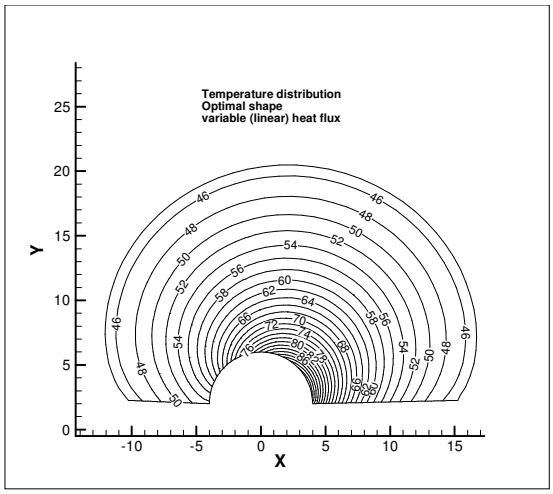
Table 3-15 Data used for Test case 7 (CGM, variable heat flux).



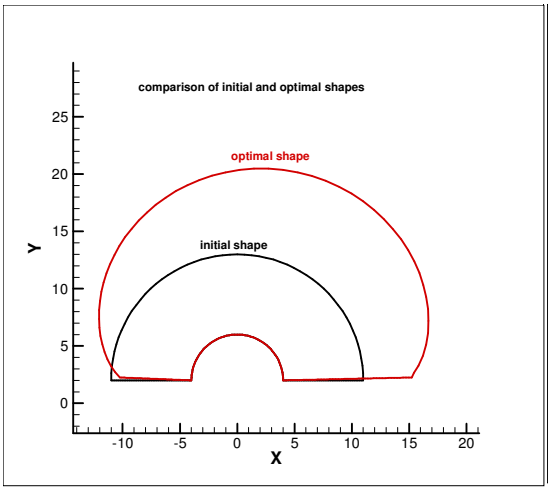
a) b)


Temperature distribution
Optimal shape
variable (linear) heat flux
T
90
85
80
75
70
65
60
55
50
Y
X

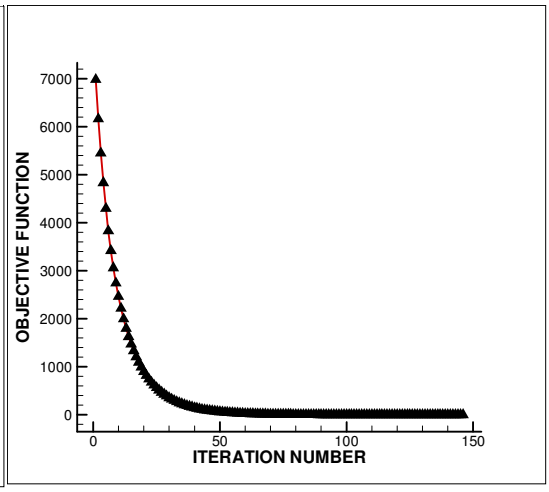
c)


Temperature distribution
Optimal shape
variable (linear) heat flux
Y
X

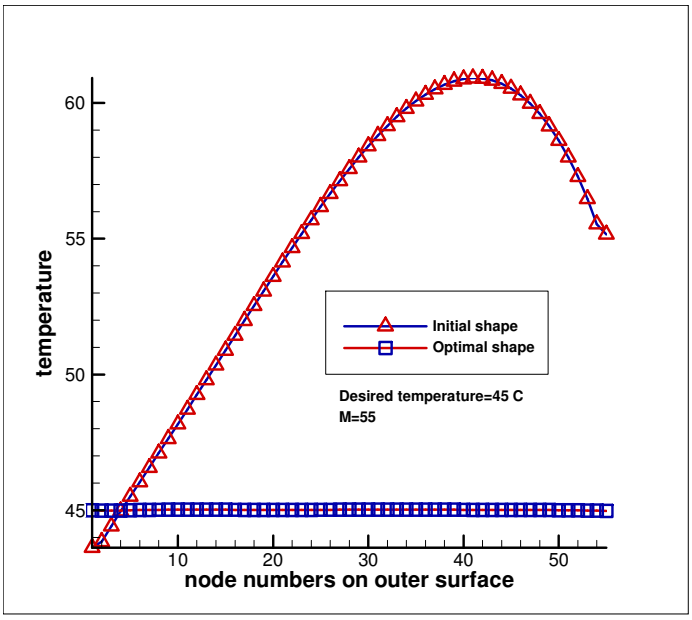
d)


comparison of initial and optimal shapes
optimal shape
initial shape
Y
X

e)


OBJECTIVE FUNCTION
ITERATION NUMBER

f)



g)

Figure 3-19 Temperature distribution for the initial shape [(a),(b)] and the optimal shape [(c),(d)], comparison of the initial and optimal shapes (e), objective function versus iteration number (f), and the comparison of the temperatures for the outer surface nodes for both initial and optimal shapes (g).

Noting that the heat flux increases linearly on the small semicircle (Figure 3-19a), the development of the outer boundary (large semicircle) away from the boundary involving the heat flux (Figure 3-19e) corresponds to the increase in the heat flux. The summary of the results is given in Table 3-16.

Decrease in objective function	From 6979.07 to 0.01
Percentage of decrease in objective function	%100
Computation time	7 min : 50 seconds
Number of iterations	146

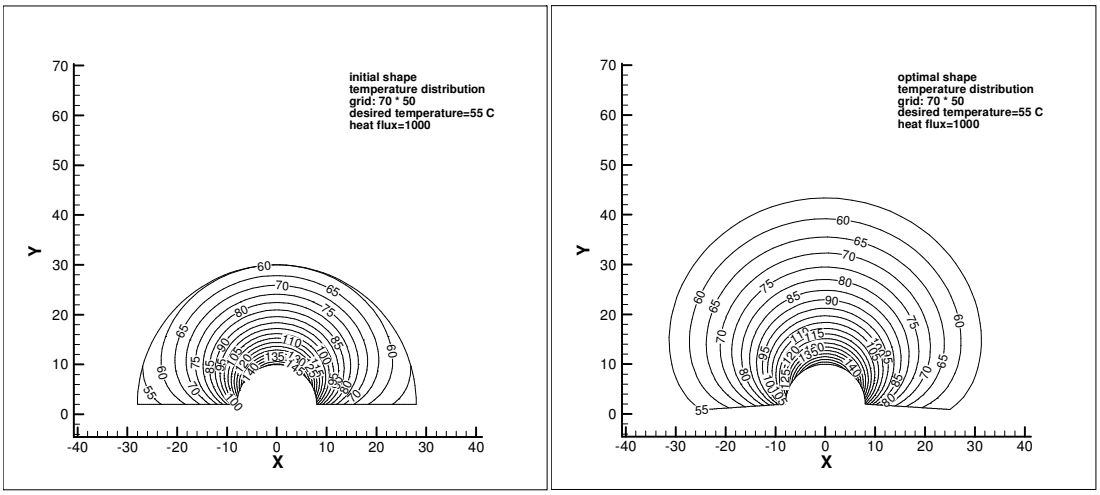
Table 3-16 Summary of results for Test case 7 (CGM, variable heat flux).

Test case 8 (CGM): Different heat fluxes

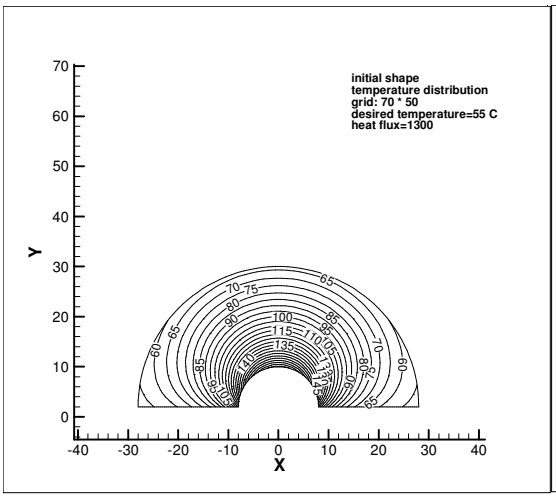
In this test case, three different values for heat flux (1000, 1300, and 1500 $(\frac{\mathrm{W}}{\mathrm{m}^2})$) are considered. All other parameters are held constant. The data used for this test case is given in Table 3-17.

$\dot{q}_1 = 1000,1300,1500(\frac{\mathrm{W}}{\mathrm{m}^2}), h_2 = 20(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_3 = 20(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_4 = 20(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}})$
$k = 90(\frac{\mathrm{W}}{\mathrm{m.C}}), T_{\infty_2} = 50(^{\circ}\mathrm{C}), T_{\infty_3} = 50(^{\circ}\mathrm{C}), T_{\infty_4} = 50(^{\circ}\mathrm{C}), T_d = 55(^{\circ}\mathrm{C})$

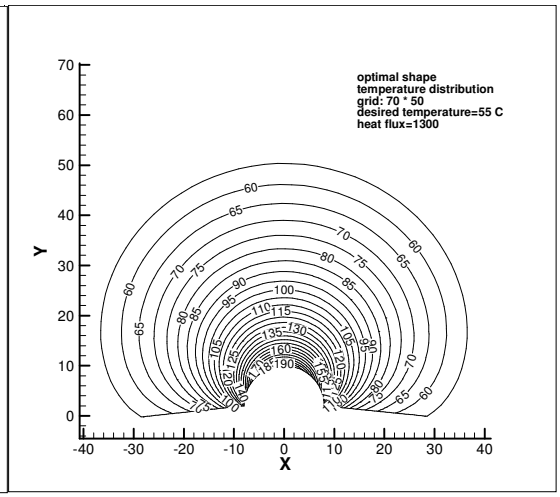
Table 3-17 Data used for Test case 8 (CGM)



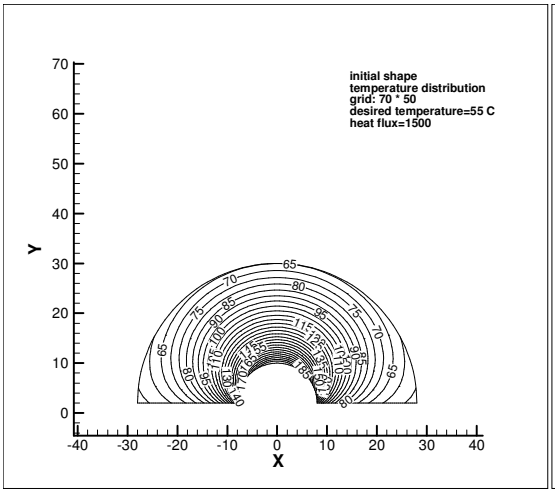
a) b)


initial shape
temperature distribution
grid: 70 * 50
desired temperature=55 C
heat flux=1300
Y
X

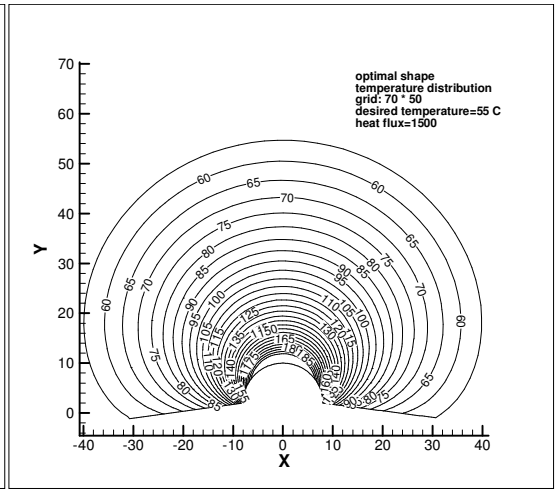
c)


optimal shape
temperature distribution
grid: 70 * 50
desired temperature=55 C
heat flux=1300
Y
X

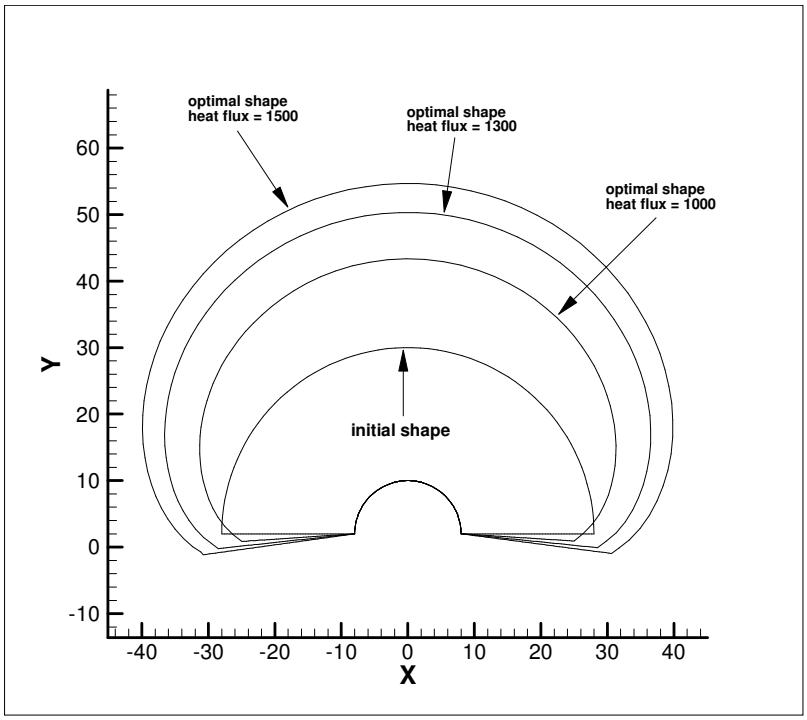
d)


initial shape
temperature distribution
grid: 70 * 50
desired temperature=55 C
heat flux=1500
Y
X

e)


optimal shape
temperature distribution
grid: 70 * 50
desired temperature=55 C
heat flux=1500
Y
X

f)



g)

Figure 3-20 Using three different heat fluxes for a problem. The initial and optimal shapes are compared for each heat flux $\dot{q}$ [(a), (b) for $\dot{q}_1 = 1000(\frac{\mathrm{W}}{\mathrm{m}^2})$, (c), (d) for $\dot{q}_1 = 1300(\frac{\mathrm{W}}{\mathrm{m}^2})$, and (e), (f) for $\dot{q}_1 = 1500(\frac{\mathrm{W}}{\mathrm{m}^2})$]. Comparison of all optimal shapes and initial shape is shown in (g).

	$\dot{q}_1 = 1000(\frac{\mathrm{W}}{\mathrm{m}^2})$	$\dot{q}_1 = 1300(\frac{\mathrm{W}}{\mathrm{m}^2})$	$\dot{q}_1 = 1500(\frac{\mathrm{W}}{\mathrm{m}^2})$
Final objective function	0.009	0.035	0.040

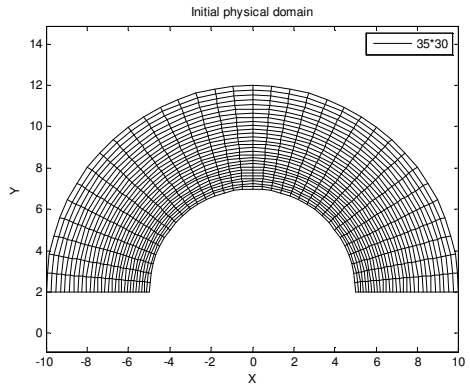
Table 3-18 Summary of results for Test case 8 (CGM).

Test case 9 (CGM): Different grid sizes & initial shapes

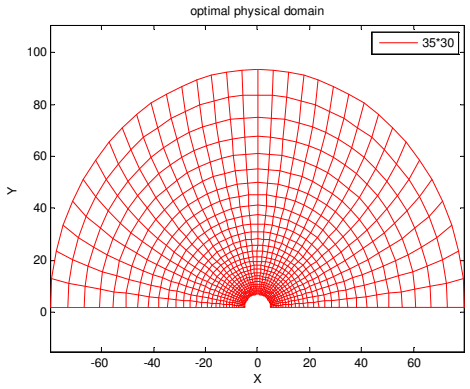
To investigate the effect of the grid size as well as initial shape on the optimization performance, the following test case is considered where the optimal shape design is inferred using the elliptic grid generation and the CGM as optimization method. Three different grid sizes 35×30, 50×40, and 60×50 are employed to study the effect of the grid size. Moreover, three different initial shapes are considered to study the effect of the initial shape. The data used for this test case is given in Table 3-19.

$\dot{q}_1 = 500(\frac{\mathrm{W}}{\mathrm{m}^2}), h_2 = 5(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_3 = 5(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_4 = 5(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}})$
$k = 20(\frac{\mathrm{W}}{\mathrm{m.C}}), T_{\infty_2} = 30(^{\circ}\mathrm{C}), T_{\infty_3} = 30(^{\circ}\mathrm{C}), T_{\infty_4} = 30(^{\circ}\mathrm{C}), T_d = 31(^{\circ}\mathrm{C})$

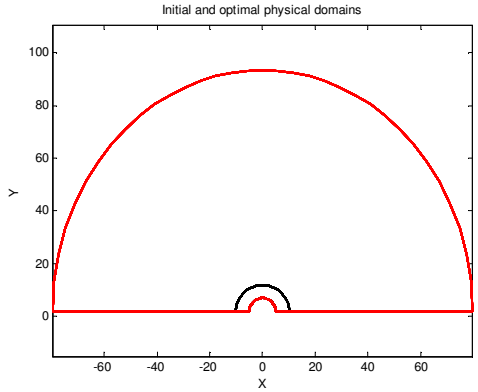
Table 3-19 Data used for Test case 9 (CGM, different grid sizes and initial shapes)



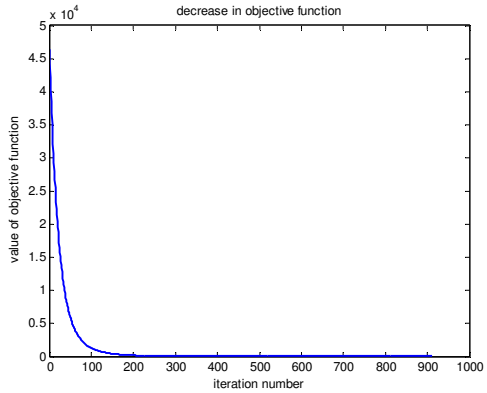
a)



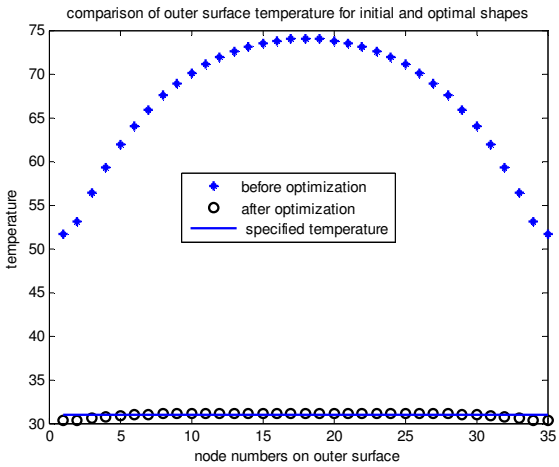
b)



c)

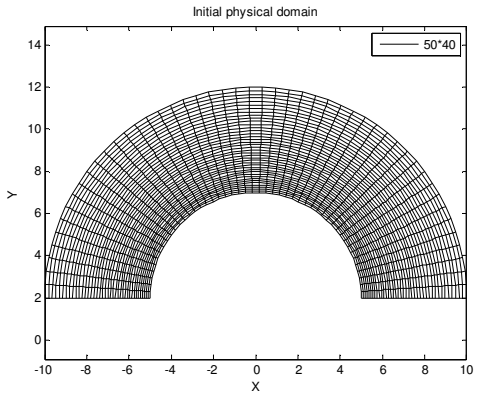


d)

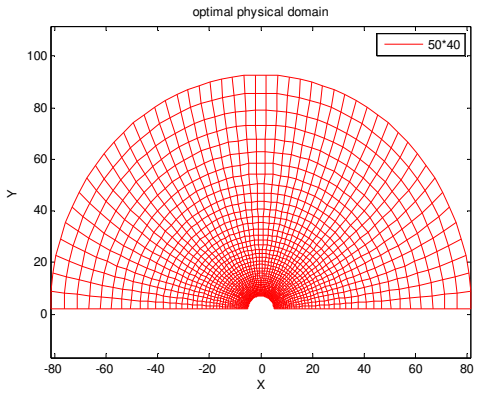


e)

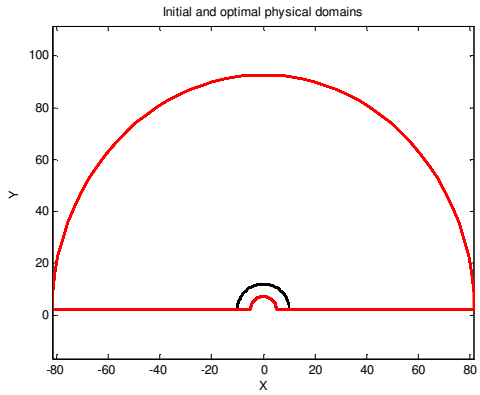
Figure 3-21 Results when using grid size 35×30 .



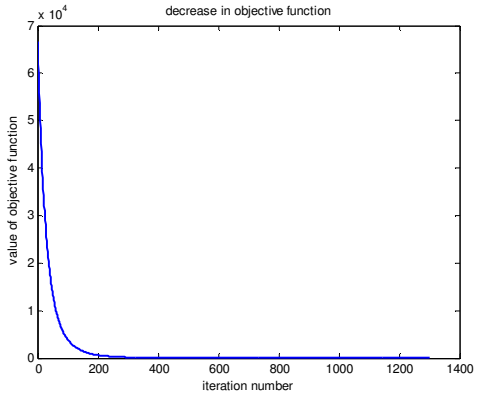
a)



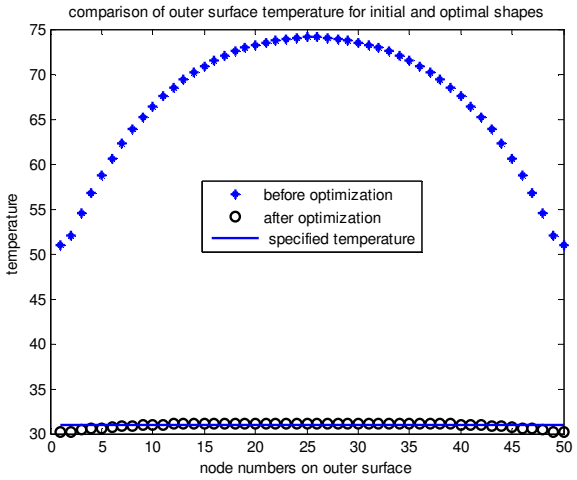
b)



c)

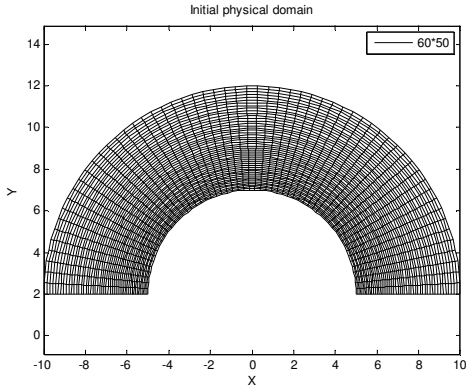


d)

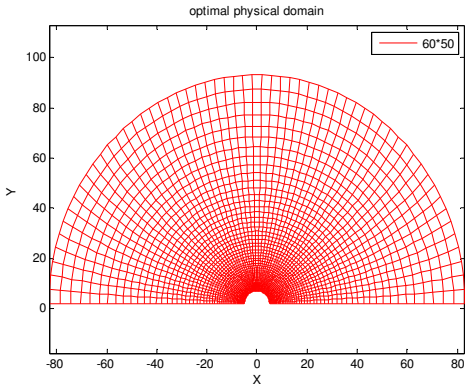


e)

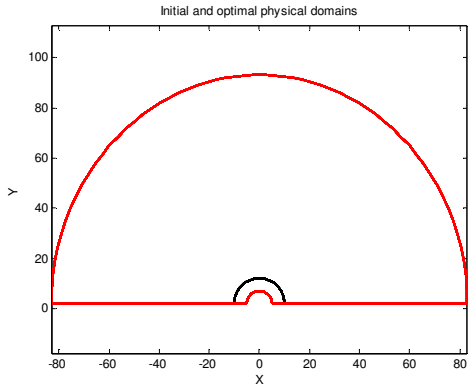
Figure 3-22 Results when using grid size 50×40.



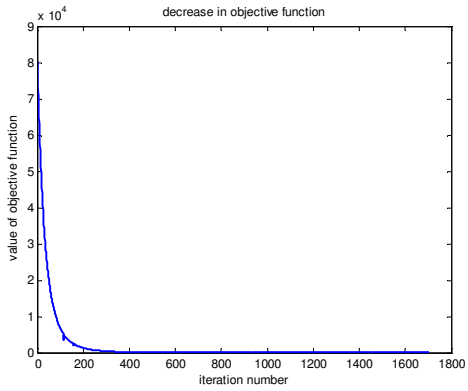
a)



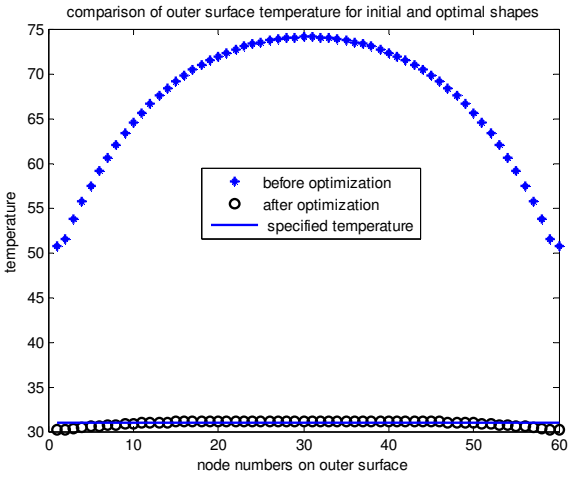
b)



c)



d)



e)

Figure 3-23 Results when using grid size 60×50.

	Grid size 35×30	Grid size 50×40	Grid size 60×50
Decrease in objective function	From 46230.4 to 2.4	From 66627.2 to 4.4	From 80207.3 to 5.7
Number of iterations	910	1300	1700
Percentage of decrease in objective function	%100	%100	%100

Table 3-20 Summary of results for three different grid sizes

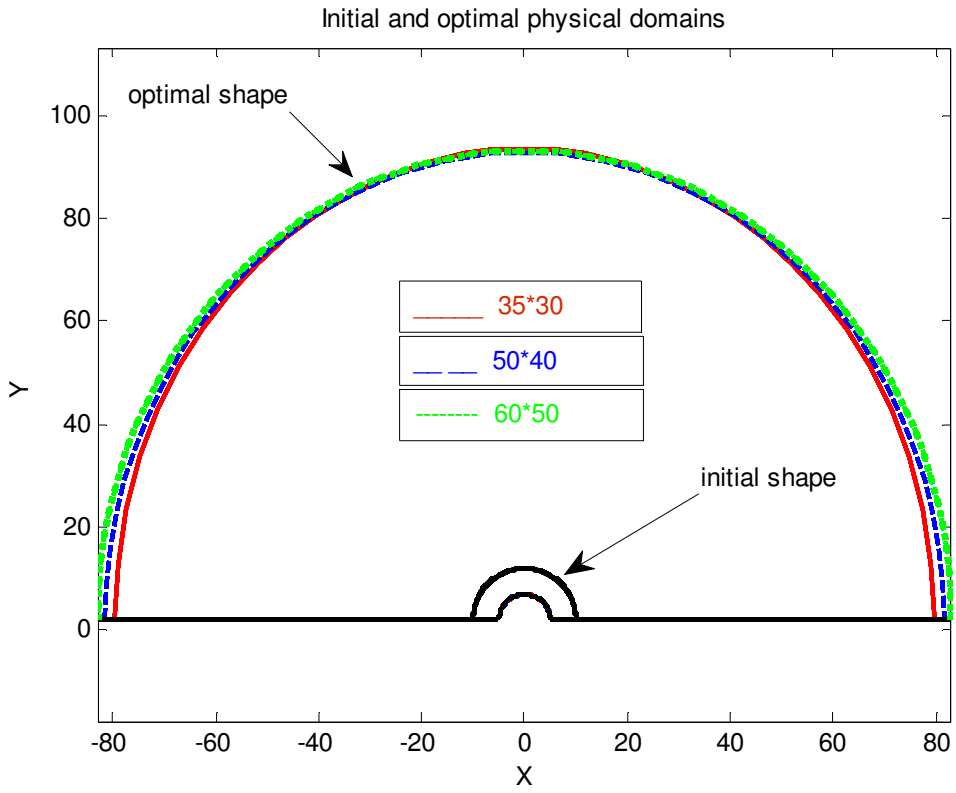


Figure 3-24 Comparison of the initial and optimal shapes for three different grid sizes. The plot shows that the optimal shape is approximately independent of the grid size.

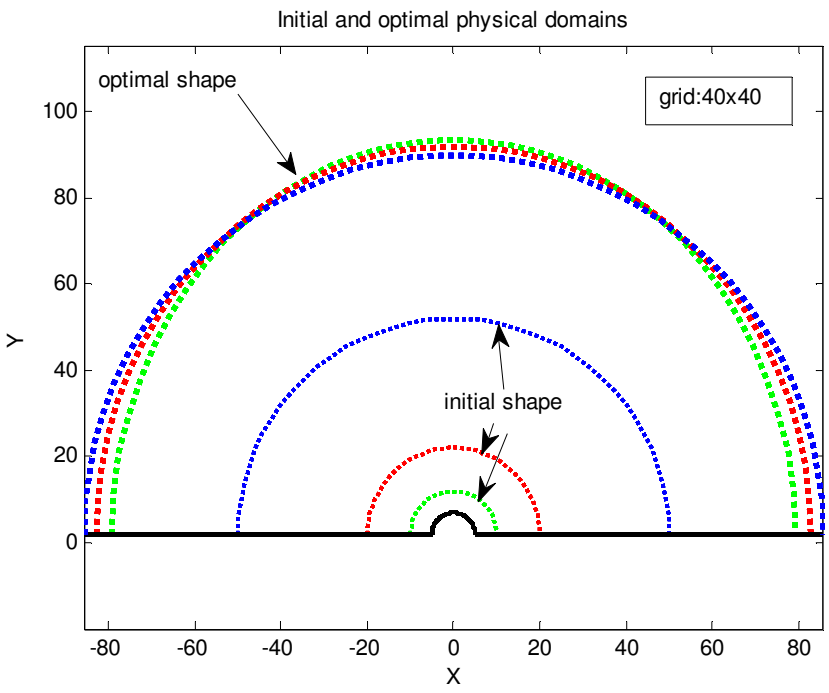


Figure 3-25 Comparison of the initial and optimal shapes for three different initial shapes. The plot shows that the optimal shape is approximately independent of the initial shape.

Comparison of LMA and CGM: With regard to the results from using the LMA (Test cases 1 to 3) and CGM (Test cases 4 to 9) in the shape optimization problems considered in this chapter, it can be concluded that the CGM is more effective than the LMA. The following is the comparison of the CGM and LMA:

- In contrast to the LMA, the CGM does not need the storage of a $n \times n$ matrix.
- The efficiency of the optimization using the CGM is higher than LMA. The temperatures for the boundary nodes of the physical domain fully matched the desired ones with the CGM only.
- In LMA, the use of fine grids to obtain a smooth and accurate shape may be time consuming and of high computational cost due to the matrix inversion in the LMA algorithm.

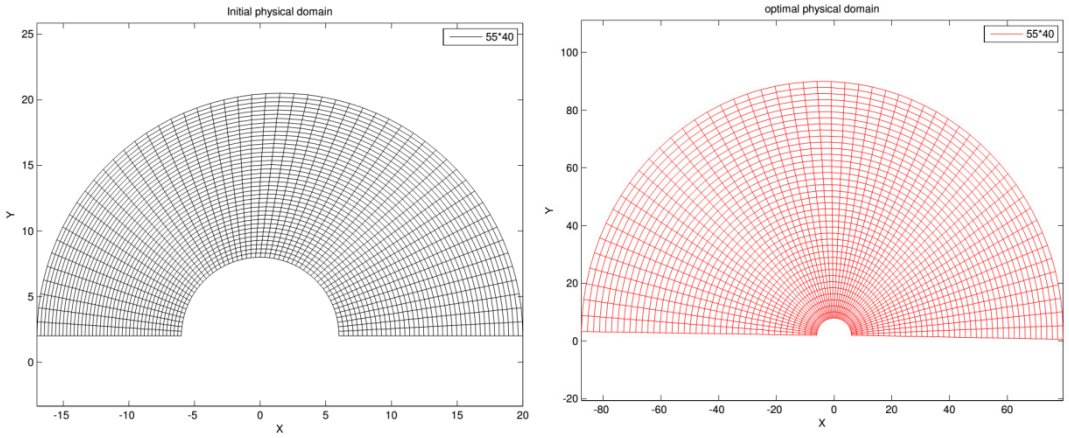
TFI and CGM: Using the TFI grid generation method in the shape optimization in heat transfer also reveals an excellent performance. However,

if there is an oscillation on the outer surface during the optimization process, the oscillation propagates into the interior of the physical domain which may result in the termination of the iterative process. For evaluating the TFI grid generation method performance in our shape optimization problems, consider the following heat transfer problems.

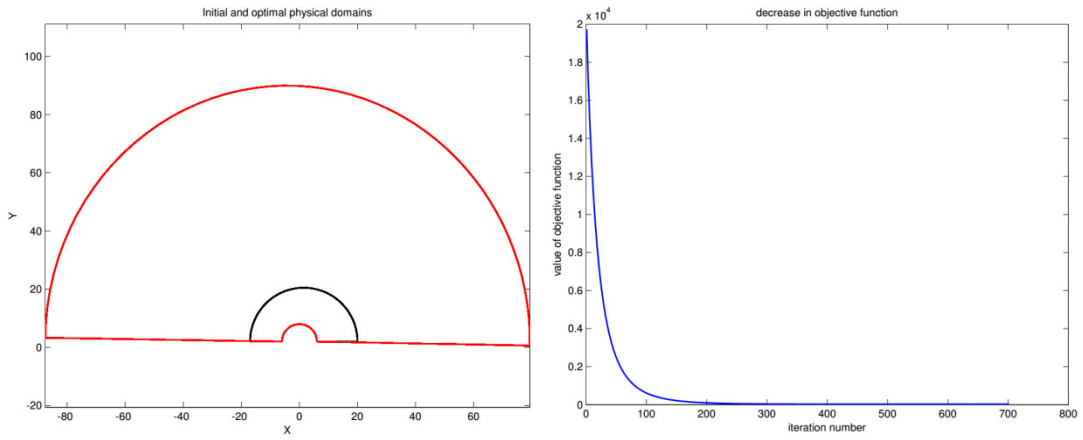
Test case 10 (CGM): using the TFI grid generation method

$\dot{q}_1 = 400(\frac{\mathrm{W}}{\mathrm{m}^2}), k = 40(\frac{\mathrm{W}}{\mathrm{m.C}}), h_2 = 4(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_3 = 4(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}}), h_4 = 4(\frac{\mathrm{W}}{\mathrm{m}^2.\mathrm{C}})$
$T_{\infty_2} = 40(^{\circ}\mathrm{C}), T_{\infty_3} = 40(^{\circ}\mathrm{C}), T_{\infty_4} = 40(^{\circ}\mathrm{C}), T_d = 42(^{\circ}\mathrm{C})$

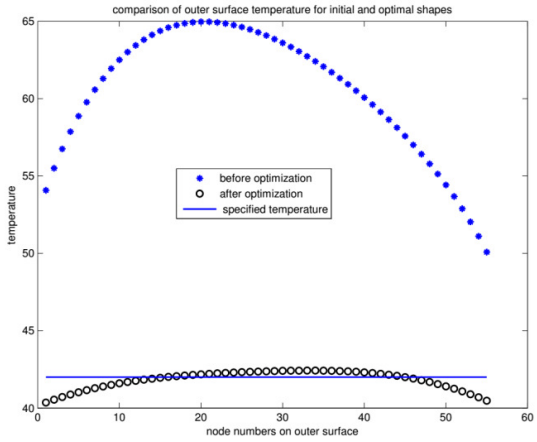
Table 3-21 Data used for Test case 10 (CGM, TFI)



a) b)



c) d)



e)

Figure 3-26 The optimal shape design based on TFI grid generation method and the conjugate gradient method.

A summary of the results is given in Table 3-22. It shows a good performance to decrease the objective function smoothly. As mentioned before, however, any probable oscillation in position of nodes on the outer surface propagates into the interior of the physical domain. This problem was not observed in the shape optimization problems using the elliptic grid generation method due to the inherent smoothness in the grids generated using this method. From Figure 3-2 (Redistribution method Section), it can be seen that the grids in the physical domain were smooth despite the ill-ordering of the outer surface nodes. Therefore, the elliptic grid generation combined with the conjugate gradient method is the optimal shape optimization method in our heat

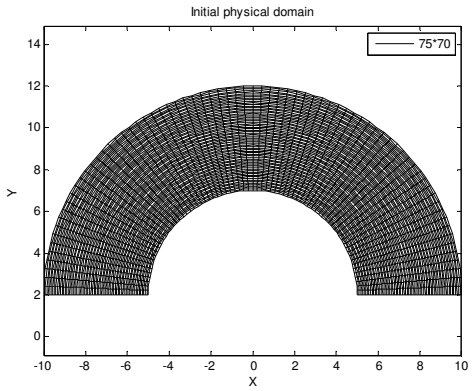
transfer problems. The 3-D shape optimization in heat transfer problems considered in this study will be processed with this approach.

Decrease in objective function	From 19722.40 to 21.18
Percentage of decrease in objective function	%99.9
Computation time	19 min
Number of iterations	701

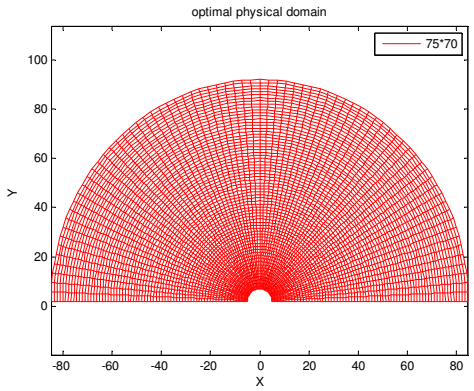
Table 3-22 Summary of results for Test case 10 (CGM, TFI)

Test case 11 (CGM): using the TFI grid generation method

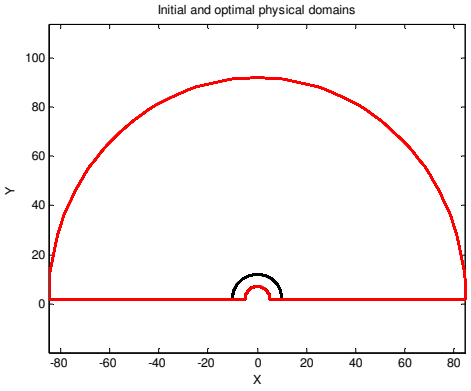
Test case 9 is repeated here but using TFI grid generation instead of elliptic one. Therefore, the data used here is the same as ones used for Test case 9. Grid size is 75×70.



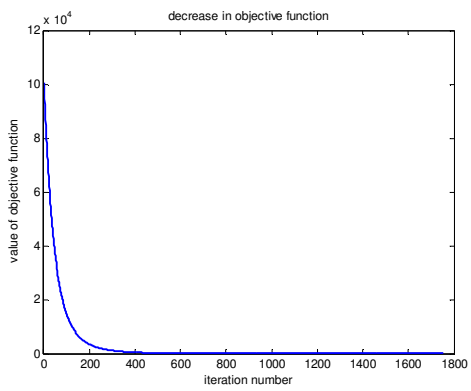
a)



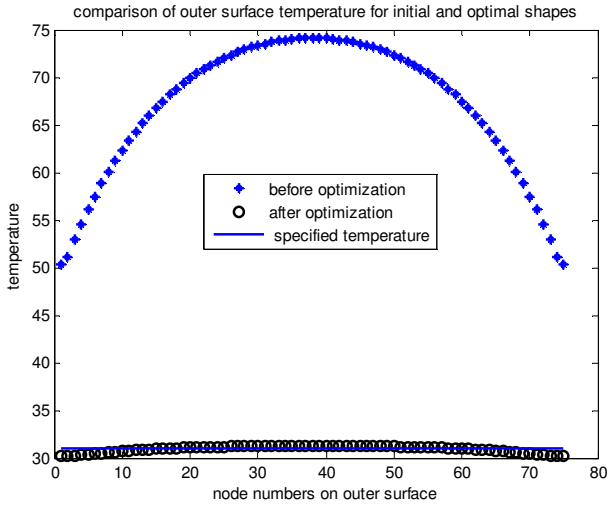
b)



c)



d)



e)

Figure 3-27 The optimal shape design based on TFI grid generation method and the conjugate gradient method.

Decrease in objective function	From 100613.80 to 9.41
Percentage of decrease in objective function	%100
Computation time	2h:7min
Number of iterations	1750

Table 3-23 Summary of results for Test case 11 (CGM, TFI)

3.9 Closing remarks

The next chapters will be devoted to the development of the procedures used in the Chapters 2 and 3. The development consists of the implementation of the three-dimensional elliptic grid generation technique and the associated field solver (the Laplace equation) for the heat conduction problem, and the sensitivity analysis and shape optimization for a three-dimensional heat conduction problem under given boundary conditions. Furthermore, aerodynamic shape optimization of an airfoil will be addressed

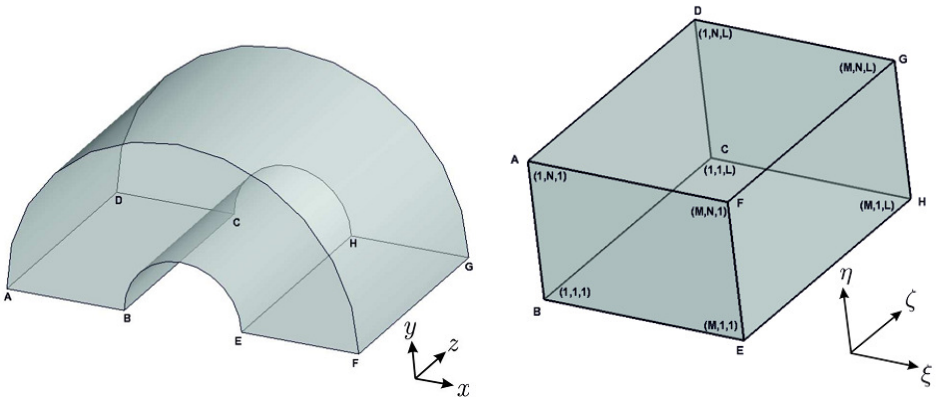
in Chapter 6. To achieve this, the formulation and the boundary conditions of velocity potential or potential flows will be studied. Since the potential flows are governed by Laplace's equation, the implementation of the Laplace's equation for the heat conduction problem investigated in Chapter 2 can be used for the potential flow equation. The objective function for this case will be defined and the sensitivity analysis and optimization procedure will be studied in details.

4. Three Dimensional Grid Generation and Direct Solver

4.1 Introduction

To apply the optimal shape design procedure we have so far used to three-dimensional problems in heat transfer, we first need a mathematical description of the three-dimensional elliptic grid generation method. The mathematical expressions for the two-dimensional elliptic grid generation technique were given in Chapter 2. In this chapter, the formulation for three-dimensional elliptic grid generation method will be developed. The equations for the three-dimensional direct solver with the associated boundary conditions will be given afterward. Both Neumann and Robin boundary conditions and the approach to implement them into the direct solver will be studied. The last section of the chapter will be devoted to the validation of the results from applying the 3-D elliptic grid generation method and direct solver with the results obtained from using the recognized FEM package ANSYS.

By solving numerically the three-dimensional Laplace or Poisson equation, one can derive the 3-D transformation relations needed to map the irregular physical domain from the x, y, and z physical space onto the cuboid ξ, η, and ζ computational domain (See Figure 4-1).



a) Physical domain b) Computational domain

Figure 4-1 The irregular physical domain and the corresponding regular computational one. The physical domain is mapped onto the computational one to solve the governing PDE for the problem.

All numerical computations are to be performed in the regular computational domain. Therefore, the grid generation problem is the determination of the unknowns (x,y,z) of the physical domain corresponding to the known (ξ,η,ζ) grid locations in the computational domain. In order to find relations to transform a PDE from the independent variables x, y, and z to the independent variables ξ, η, and ζ, the following equations may be used [76]

$$x \equiv x(\xi,\eta,\zeta) \tag{4.1}$$

$$y \equiv y(\xi,\eta,\zeta) \tag{4.2}$$

$$z \equiv z(\xi,\eta,\zeta) \tag{4.3}$$

and the inverse transformation is given below

$$\xi \equiv \xi(x,y,z) \tag{4.4}$$

$$\eta \equiv \eta(x,y,z) \tag{4.5}$$

$$\zeta \equiv \zeta(x,y,z) \tag{4.6}$$

The three elliptic PDEs of the Poisson type employed to derive such a transformation are given by

$$\xi_{xx} + \xi_{yy} + \xi_{zz} = P(\xi, \eta, \zeta) \tag{4.7}$$

$$\eta_{xx} + \eta_{yy} + \eta_{zz} = Q(\xi, \eta, \zeta) \tag{4.8}$$

$$\zeta_{xx} + \zeta_{yy} + \zeta_{zz} = R(\xi, \eta, \zeta) \tag{4.9}$$

where ξ, η, and ζ are the computational coordinates corresponding to x, y, and z in the physical coordinate, respectively. P, Q, and R are grid control functions which control the density of nodes towards a specified grid line or about a specified grid point. Since Laplace or Poisson equation involves second derivatives, relationships are needed to transform such derivatives from the (x,y,z) system to the (ξ,η,ζ) one. To perform this, the Jacobian of the transformation is needed which is given below

$$J = J(\frac{x,y,z}{\xi,\eta,\zeta})$$

$$= \begin{vmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & z_\zeta \end{vmatrix} = x_\xi y_\eta z_\zeta - x_\xi y_\zeta z_\eta + x_\eta y_\zeta z_\xi - x_\eta y_\xi z_\zeta + x_\zeta y_\xi z_\eta - x_\zeta y_\eta z_\xi \tag{4.10}$$

The following three elliptic PDEs may be solved in the computational domain to determine the unknowns x, y, and z values of the physical domain [102]

$$\alpha_{11} x_{\xi\xi} + \alpha_{22} x_{\eta\eta} + \alpha_{33} x_{\zeta\zeta} + 2(\alpha_{12} x_{\xi\eta} + \alpha_{13} x_{\xi\zeta} + \alpha_{23} x_{\eta\zeta}) = -J^2(P x_\xi + Q x_\eta + R x_\zeta)$$

(4.11)

$$\alpha_{11} y_{\xi\xi} + \alpha_{22} y_{\eta\eta} + \alpha_{33} y_{\zeta\zeta} + 2(\alpha_{12} y_{\xi\eta} + \alpha_{13} y_{\xi\zeta} + \alpha_{23} y_{\eta\zeta}) = -J^2(P y_\xi + Q y_\eta + R y_\zeta)$$

(4.12)

$$\alpha_{11} z_{\xi\xi} + \alpha_{22} z_{\eta\eta} + \alpha_{33} z_{\zeta\zeta} + 2(\alpha_{12} z_{\xi\eta} + \alpha_{13} z_{\xi\zeta} + \alpha_{23} z_{\eta\zeta}) = -J^2(P z_\xi + Q z_\eta + R z_\zeta) \tag{4.13}$$

where J denotes the Jacobian of transformation. The metric coefficients α are computed by the following relations

$$\alpha_{11} = (x_\eta^2 + y_\eta^2 + z_\eta^2)(x_\zeta^2 + y_\zeta^2 + z_\zeta^2) - (x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta)^2 \qquad (4.14)$$

$$\alpha_{22} = (x_\xi^2 + y_\xi^2 + z_\xi^2)(x_\zeta^2 + y_\zeta^2 + z_\zeta^2) - (x_\xi x_\zeta + y_\xi y_\zeta + z_\xi z_\zeta)^2 \qquad (4.15)$$

$$\alpha_{33} = (x_\xi^2 + y_\xi^2 + z_\xi^2)(x_\eta^2 + y_\eta^2 + z_\eta^2) - (x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta)^2 \qquad (4.16)$$

$$\alpha_{12} = (x_\xi x_\zeta + y_\xi y_\zeta + z_\xi z_\zeta)(x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta) - (x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta)(x_\zeta^2 + y_\zeta^2 + z_\zeta^2) \qquad (4.17)$$

$$\alpha_{13} = (x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta)(x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta) - (x_\xi x_\zeta + y_\xi y_\zeta + z_\xi z_\zeta)(x_\eta^2 + y_\eta^2 + z_\eta^2) \qquad (4.18)$$

$$\alpha_{23} = (x_\xi x_\zeta + y_\xi y_\zeta + z_\xi z_\zeta)(x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta) - (x_\eta x_\zeta + y_\eta y_\zeta + z_\eta z_\zeta)(x_\xi^2 + y_\xi^2 + z_\xi^2) \qquad (4.19)$$

The numerical solution of the above elliptic PDEs yields the explicit relations for x, y, and z in terms of grid points $\xi_i (i \in [1, M])$, $\eta_j (j \in [1, N])$, and $\zeta_k (k \in [1, L])$ in the computational domain. By setting the grid control functions to zero $(P = Q = R = 0)$, the Poisson equations reduce to Laplace ones. The x, y, and z coordinates of the grid points at the corners (e.g. the points A, B, C, D, E, F, G, and H in Figure 4-1) of the computational domain remain constant and are equal to corresponding ones in the physical domain. Therefore, for $f \equiv x$, y, or z

$$f_{1,1,1}\,|_{\text{comp.domain}} = f_B\,|_{\text{phys.domain}}, f_{M,1,1}\,|_{\text{comp.domain}} = f_E\,|_{\text{phys.domain}} \qquad (4.20)$$

$$f_{1,N,1}\,|_{\text{comp.domain}} = f_A\,|_{\text{phys.domain}}, f_{M,N,1}\,|_{\text{comp.domain}} = f_F\,|_{\text{phys.domain}} \qquad (4.21)$$

$$f_{1,1,L}\,|_{\text{comp.domain}} = f_C\,|_{\text{phys.domain}}, f_{M,1,L}\,|_{\text{comp.domain}} = f_H\,|_{\text{phys.domain}} \qquad (4.22)$$

$$f_{1,N,L}\,|_{\text{comp.domain}} = f_D\,|_{\text{phys.domain}}, f_{M,N,L}\,|_{\text{comp.domain}} = f_G\,|_{\text{phys.domain}} \qquad (4.23)$$

Twelve cuboid sides BE, AF, CH, DG, BC, EH, AD, FG, BA, EF, CD, HG may be discretized uniformly by imposing $(M-1)$ equal intervals on sides BE, AF, CH, and DG, $(N-1)$ equal intervals on sides BA, EF, CD, and HG, and $(L-1)$ equal intervals on sides BC, EH, AD, and FG. Afterwards, for generating grid over boundary faces $\Gamma_i (i=1,\ldots,6)$, we can simply apply a 2-D algebraic grid generation technique such as TFI (TransFinite Interpolation) or the 2-D elliptic grid generation one. In this thesis, both TFI and elliptic grid generation methods are used depending on the shape of boundary face.

Using the finite difference method, Equation (4.10) and Equations (4.14) through (4.19) can be discretized. Assuming

$$\Delta\xi = \Delta\eta = \Delta\zeta = 1 \tag{4.24}$$

we have

$$(f_\xi)_{i,j,k} = \frac{1}{2}(f_{i+1,j,k} - f_{i-1,j,k}) \tag{4.25}$$

$$(f_\eta)_{i,j,k} = \frac{1}{2}(f_{i,j+1,k} - f_{i,j-1,k}) \tag{4.26}$$

$$(f_\zeta)_{i,j,k} = \frac{1}{2}(f_{i,j,k+1} - f_{i,j,k-1}) \tag{4.27}$$

$$(f_{\xi\xi})_{i,j,k} = (f_{i+1,j,k} - 2f_{i,j,k} + f_{i-1,j,k}) \tag{4.28}$$

$$(f_{\eta\eta})_{i,j,k} = (f_{i,j+1,k} - 2f_{i,j,k} + f_{i,j-1,k}) \tag{4.29}$$

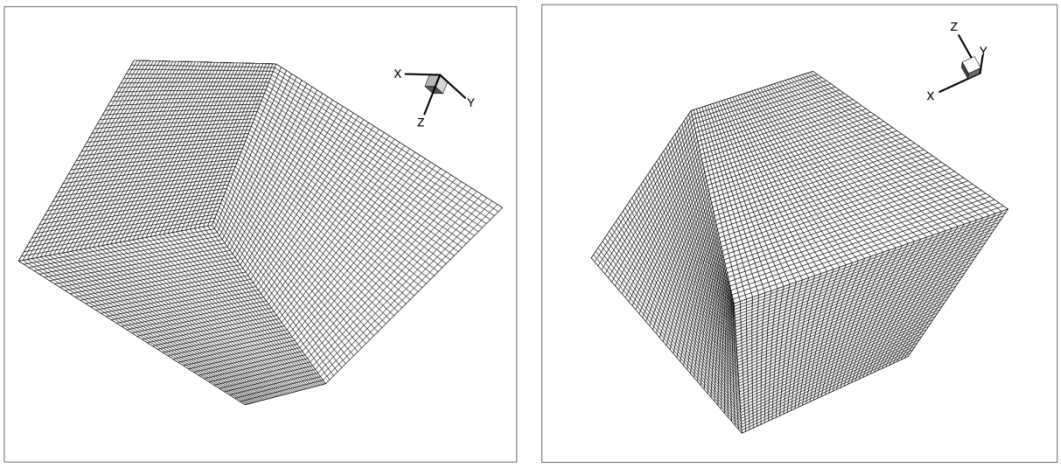
$$(f_{\zeta\zeta})_{i,j,k} = (f_{i,j,k+1} - 2f_{i,j,k} + f_{i,j,k-1}) \tag{4.30}$$

$$(f_{\xi\eta})_{i,j,k} = \frac{1}{4}(f_{i+1,j+1,k} - f_{i-1,j+1,k} - f_{i+1,j-1,k} + f_{i-1,j-1,k}) \tag{4.31}$$

$$(f_{\xi\zeta})_{i,j,k} = \frac{1}{4}(f_{i+1,j,k+1} - f_{i-1,j,k+1} - f_{i+1,j,k-1} + f_{i-1,j,k-1}) \tag{4.32}$$

$$(f_{\eta\zeta})_{i,j,k} = \frac{1}{4}(f_{i,j+1,k+1} - f_{i,j-1,k+1} - f_{i,j+1,k-1} + f_{i,j-1,k-1}) \qquad (4.33)$$

The relationships above are valid for all interior nodes, i.e. $i \in [2, M-1]$, $j \in [2, N-1]$, and $k \in [2, L-1]$ where M, N, and L are the number of nodes in the ξ, η, and ζ directions, respectively. A non-linear set of algebraic equations is then obtained in terms of $x_{i,j,k}$, $y_{i,j,k}$, and $z_{i,j,k}$ by substituting the above finite difference relationships into Equation (4.10) and Equations (4.14) through (4.19), substituting the metric coefficients α_{11}, α_{12}, ..., α_{33} and the Jacobian of the transformation into Equations (4.11) to (4.13), and assuming $(P = Q = R = 0)$ for generating a smooth grid over the physical domain. In order to solve these equations, the Successive Over Relaxation (SOR) is used due to its high convergence rate relative to other iterative methods such as the Gauss-Seidel method. Several examples of elliptic grid generation for two different geometries are shown in Figure 4-2. The grid size is $45 \times 45 \times 45$.



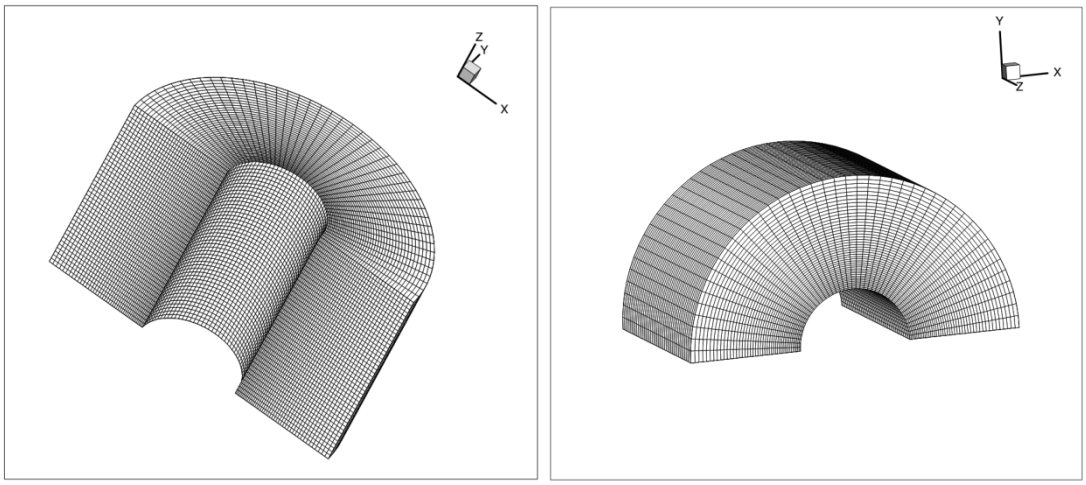


Figure 4-2 Grids for two different geometries generated by the elliptic grid generation technique. The grid size is $45 \times 45 \times 45$.

4.2 Field Equation Solver

The 3-D elliptic grid generation technique and the 3-D steady heat conduction equation both satisfy the Laplace's equation and therefore much of the effort put into programming the relationships required to generate the grid over the physical domain can be reused to solve the heat transfer problem. Consider the 3-D steady state heat conduction equation

$$T_{xx} + T_{yy} + T_{zz} = 0 \tag{4.34}$$

in the physical domain shown in Figure 4-3 subject to a combination of Neumann and Robin boundary conditions.

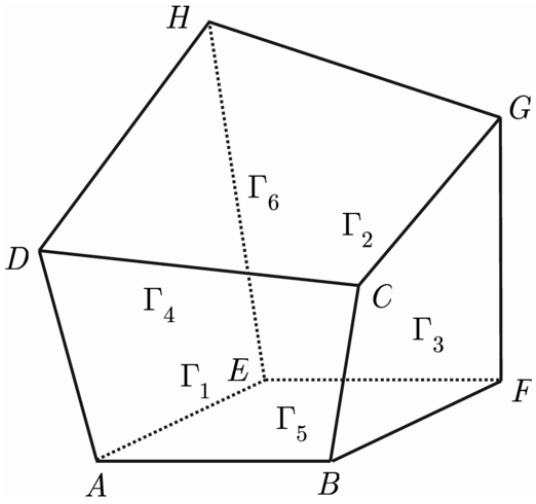


Figure 4-3 arbitrarily shaped heat conducting body under specified boundary conditions.

The surface Γ_5 is subject to a prescribed heat flux $\dot{q}$, and the surfaces $\Gamma_i (i = 1,2,3,4,6)$ are exposed to convective heat transfer with the corresponding heat transfer coefficients $h_i \left(i = 1,2,3,4,6\right)$ and surrounding temperatures $T_{\infty_i} (i = 1,2,3,4,6)$, respectively. The heat conduction equation and the corresponding boundary conditions must be transformed from the x, y, and z physical coordinates to the ξ, η, and ζ computational one. The transformation results in

$$\alpha_{11} T_{\xi\xi} + \alpha_{22} T_{\eta\eta} + \alpha_{33} T_{\zeta\zeta} + 2(\alpha_{12} T_{\xi\eta} + \alpha_{13} T_{\xi\zeta} + \alpha_{23} T_{\eta\zeta}) = -J^2 (P T_\xi + Q T_\eta + R T_\zeta) \tag{4.35}$$

where the metric coefficients α_{11}, α_{12}, ..., α_{33} and the Jacobian of the transformation J are defined by Equations (4.14) through (4.19) and (4.10), respectively. The boundary conditions frequently encountered in the heat transfer are the Dirichlet, Neumann, and Robin types. If the boundary conditions are of Dirichlet kind (such as temperature distribution), no transformation is required as the temperature values remain constant in the computational domain. However, if there are Neumann conditions (such as heat flux on a boundary) or Robin conditions (such as convective heat transfer on a boundary), a transformation is required as the temperature derivative (in heat flux) in the physical domain is not the same as that in the

computational domain. In fact, $\frac{\partial T}{\partial n}$ at a boundary in the physical domain is related to $\frac{\partial T}{\partial \xi}$, $\frac{\partial T}{\partial \eta}$, and $\frac{\partial T}{\partial \zeta}$ in the computational one (see Figure 4-4).

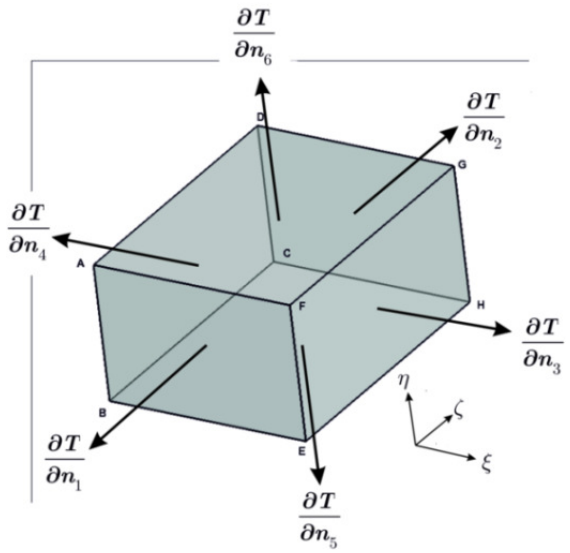


Figure 4-4 Normal derivatives $\frac{\partial T}{\partial n_i}(i = 1,2,\ldots,6)$ along the outward drawn unit vectors.

The use of chain rule to derive such a transformation gives

$$\frac{\partial T}{\partial x} = \frac{\partial T}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial T}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial T}{\partial \zeta}\frac{\partial \zeta}{\partial x} \tag{4.36}$$

$$\frac{\partial T}{\partial y} = \frac{\partial T}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial T}{\partial \eta}\frac{\partial \eta}{\partial y} + \frac{\partial T}{\partial \zeta}\frac{\partial \zeta}{\partial y} \tag{4.37}$$

$$\frac{\partial T}{\partial z} = \frac{\partial T}{\partial \xi}\frac{\partial \xi}{\partial z} + \frac{\partial T}{\partial \eta}\frac{\partial \eta}{\partial z} + \frac{\partial T}{\partial \zeta}\frac{\partial \zeta}{\partial z} \tag{4.38}$$

By interchanging x and ξ, y and η, and z and ζ, the following relationships can also be derived

$$\frac{\partial T}{\partial \xi} = \frac{\partial T}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial T}{\partial y}\frac{\partial y}{\partial \xi} + \frac{\partial T}{\partial z}\frac{\partial z}{\partial \xi} \tag{4.39}$$

$$\frac{\partial T}{\partial \eta} = \frac{\partial T}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial T}{\partial y}\frac{\partial y}{\partial \eta} + \frac{\partial T}{\partial z}\frac{\partial z}{\partial \eta} \tag{4.40}$$

$$\frac{\partial T}{\partial \zeta} = \frac{\partial T}{\partial x}\frac{\partial x}{\partial \zeta} + \frac{\partial T}{\partial y}\frac{\partial y}{\partial \zeta} + \frac{\partial T}{\partial z}\frac{\partial z}{\partial \zeta} \tag{4.41}$$

or in condensed form, the equations above may be written as

$$\begin{Bmatrix} T_x x_\xi + T_y y_\xi + T_z z_\xi = T_\xi \\ T_x x_\eta + T_y y_\eta + T_z z_\eta = T_\eta \\ T_x x_\zeta + T_y y_\zeta + T_z z_\zeta = T_\zeta \end{Bmatrix} \tag{4.42}$$

The use of symbolic computation software MAPLE to solve the system of Equations (4.42) for $\frac{\partial T}{\partial x}$, $\frac{\partial T}{\partial y}$, and $\frac{\partial T}{\partial z}$ leads to

$$T_x = \frac{[(y_\eta z_\zeta - y_\zeta z_\eta)T_\xi + (y_\zeta z_\xi - y_\xi z_\zeta)T_\eta + (y_\xi z_\eta - y_\eta z_\xi)T_\zeta]}{J} \tag{4.43}$$

$$T_y = \frac{-[(x_\eta z_\zeta - x_\zeta z_\eta)T_\xi + (x_\zeta z_\xi - x_\xi z_\zeta)T_\eta + (x_\xi z_\eta - x_\eta z_\xi)T_\zeta]}{J} \tag{4.44}$$

$$T_z = \frac{[(x_\eta y_\zeta - x_\zeta y_\eta)T_\xi + (x_\zeta y_\xi - x_\xi y_\zeta)T_\eta + (x_\xi y_\eta - x_\eta y_\xi)T_\zeta]}{J} \tag{4.45}$$

By comparing Equations (4.36) to (4.38) and Equations (4.43) to (4.45), it can be concluded that

$$\xi_x = \frac{1}{J}[(y_\eta z_\zeta - y_\zeta z_\eta)] \tag{4.46}$$

$$\eta_x = \frac{1}{J}[(y_\zeta z_\xi - y_\xi z_\zeta)] \tag{4.47}$$

$$\zeta_x = \frac{1}{J}[(y_\xi z_\eta - y_\eta z_\xi)] \tag{4.48}$$

$$\xi_y = \frac{-1}{J}[(x_\eta z_\zeta - x_\zeta z_\eta)] \tag{4.49}$$

$$\eta_y = \frac{-1}{J}[(x_\zeta z_\xi - x_\xi z_\zeta)] \tag{4.50}$$

$$\zeta_y = \frac{-1}{J}[(x_\xi z_\eta - x_\eta z_\xi)] \tag{4.51}$$

$$\xi_z = \frac{1}{J}[(x_\eta y_\zeta - x_\zeta y_\eta)] \tag{4.52}$$

$$\eta_z = \frac{1}{J}[(x_\zeta y_\xi - x_\xi y_\zeta)] \tag{4.53}$$

$$\zeta_z = \frac{1}{J}[(x_\xi y_\eta - x_\eta y_\xi)] \tag{4.54}$$

Since

$$\frac{\partial T}{\partial n} = \nabla T \cdot \mathbf{n} \tag{4.55}$$

Thus,

$$\frac{\partial T}{\partial \xi} = (T_x\mathbf{i} + T_y\mathbf{j} + T_z\mathbf{k}).(\frac{\xi_x\mathbf{i} + \xi_y\mathbf{j} + \xi_z\mathbf{k}}{\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}}) = \frac{T_x\xi_x + T_y\xi_y + T_z\xi_z}{\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}} \tag{4.56}$$

which may be explicitly rewritten as

$$\begin{aligned}\frac{\partial T}{\partial \xi} = &([(y_\eta z_\zeta - y_\zeta z_\eta)T_\xi + (y_\zeta z_\xi - y_\xi z_\zeta)T_\eta + (y_\xi z_\eta - y_\eta z_\xi)T_\zeta][y_\eta z_\zeta - y_\zeta z_\eta] \\ &+[(x_\eta z_\zeta - x_\zeta z_\eta)T_\xi + (x_\zeta z_\xi - x_\xi z_\zeta)T_\eta + (x_\xi z_\eta - x_\eta z_\xi)T_\zeta][x_\eta z_\zeta - x_\zeta z_\eta] \\ &+[(x_\eta y_\zeta - x_\zeta y_\eta)T_\xi + (x_\zeta y_\xi - x_\xi y_\zeta)T_\eta + (x_\xi y_\eta - x_\eta y_\xi)T_\zeta][x_\eta y_\zeta - x_\zeta y_\eta]) \\ &/(J\sqrt{(y_\eta z_\zeta - y_\zeta z_\eta)^2 + (x_\eta z_\zeta - x_\zeta z_\eta)^2 + (x_\eta y_\zeta - x_\zeta y_\eta)^2})\end{aligned} \tag{4.57}$$

and similar expressions may be derived for $\frac{\partial T}{\partial \eta}$ and $\frac{\partial T}{\partial \zeta}$ which are as following

$$\frac{\partial T}{\partial \eta} = \frac{T_x \eta_x + T_y \eta_y + T_z \eta_z}{\sqrt{\eta_x^2 + \eta_y^2 + \eta_z^2}} \tag{4.58}$$

which may be explicitly rewritten as

$$\begin{aligned}\frac{\partial T}{\partial \eta} = &([(y_\eta z_\zeta - y_\zeta z_\eta)T_\xi + (y_\zeta z_\xi - y_\xi z_\zeta)T_\eta + (y_\xi z_\eta - y_\eta z_\xi)T_\zeta][y_\zeta z_\xi - y_\xi z_\zeta] \\ &+[(x_\eta z_\zeta - x_\zeta z_\eta)T_\xi + (x_\zeta z_\xi - x_\xi z_\zeta)T_\eta + (x_\xi z_\eta - x_\eta z_\xi)T_\zeta][x_\zeta z_\xi - x_\xi z_\zeta] \\ &+[(x_\eta y_\zeta - x_\zeta y_\eta)T_\xi + (x_\zeta y_\xi - x_\xi y_\zeta)T_\eta + (x_\xi y_\eta - x_\eta y_\xi)T_\zeta][x_\zeta y_\xi - x_\xi y_\zeta]) \\ &/(J\sqrt{(y_\zeta z_\xi - y_\xi z_\zeta)^2 + (x_\zeta z_\xi - x_\xi z_\zeta)^2 + (x_\zeta y_\xi - x_\xi y_\zeta)^2})\end{aligned} \tag{4.59}$$

and

$$\frac{\partial T}{\partial \zeta} = \frac{T_x \zeta_x + T_y \zeta_y + T_z \zeta_z}{\sqrt{\zeta_x^2 + \zeta_y^2 + \zeta_z^2}} \tag{4.60}$$

and in explicit form

$$\begin{aligned}\frac{\partial T}{\partial \zeta} = &([(y_\eta z_\zeta - y_\zeta z_\eta)T_\xi + (y_\zeta z_\xi - y_\xi z_\zeta)T_\eta + (y_\xi z_\eta - y_\eta z_\xi)T_\zeta][y_\xi z_\eta - y_\eta z_\xi] \\ &+[(x_\eta z_\zeta - x_\zeta z_\eta)T_\xi + (x_\zeta z_\xi - x_\xi z_\zeta)T_\eta + (x_\xi z_\eta - x_\eta z_\xi)T_\zeta][x_\xi z_\eta - x_\eta z_\xi] \\ &+[(x_\eta y_\zeta - x_\zeta y_\eta)T_\xi + (x_\zeta y_\xi - x_\xi y_\zeta)T_\eta + (x_\xi y_\eta - x_\eta y_\xi)T_\zeta][x_\xi y_\eta - x_\eta y_\xi]) \\ &/(J\sqrt{(y_\xi z_\eta - y_\eta z_\xi)^2 + (x_\xi z_\eta - x_\eta z_\xi)^2 + (x_\xi y_\eta - x_\eta y_\xi)^2})\end{aligned} \tag{4.61}$$

Using FDM, one can discretize the relations for $\frac{\partial T}{\partial \xi}$, $\frac{\partial T}{\partial \eta}$, and $\frac{\partial T}{\partial \zeta}$ on every boundary with Neumann and Robin conditions. For example, for nodes on surface Γ_3 with $i = M$, $j \in [2, N-1]$, and $k \in [2, L-1]$, using the *one-sided backward* and *central* finite difference relations, it can be shown that

$$\frac{\partial T}{\partial \xi} = T_{\xi} = \frac{1}{2}(3T_{M,j,k} - 4T_{M-1,j,k} + T_{M-2,j,k}) \tag{4.62}$$

$$\frac{\partial T}{\partial \eta} = T_{\eta} = \frac{1}{2}(T_{M,j+1,k} - T_{M,j-1,k}) \tag{4.63}$$

$$\frac{\partial T}{\partial \zeta} = T_{\zeta} = \frac{1}{2}(T_{M,j,k+1} - T_{M,j,k-1}) \tag{4.64}$$

Since the heat conducted in the body is dissipated to the surroundings by convection, by equating the heat conduction and convection equations on the surface Γ_3, we have

$$\dot{q}_{\text{conduction}} = \dot{q}_{\text{convection}} \tag{4.65}$$

$$-k_T \frac{\partial T}{\partial n_3} = h_3(T_{\Gamma_3} - T_{\infty_3}) \tag{4.66}$$

$$-k_T \frac{\partial T}{\partial \xi} = h_3(T_{\Gamma_3} - T_{\infty_3}) \tag{4.67}$$

By substituting the expression for $\dfrac{\partial T}{\partial \xi}$ from Equation (4.57) into the above equation, and noting that $T_{\Gamma_3} = T_{M,j,k}$, we obtain

$$\begin{aligned}
&-k_T([(y_\eta z_\zeta - y_\zeta z_\eta)T_\xi + (y_\zeta z_\xi - y_\xi z_\zeta)T_\eta + (y_\xi z_\eta - y_\eta z_\xi)T_\zeta][y_\eta z_\zeta - y_\zeta z_\eta] \\
&+[(x_\eta z_\zeta - x_\zeta z_\eta)T_\xi + (x_\zeta z_\xi - x_\xi z_\zeta)T_\eta + (x_\xi z_\eta - x_\eta z_\xi)T_\zeta][x_\eta z_\zeta - x_\zeta z_\eta] \\
&+[(x_\eta y_\zeta - x_\zeta y_\eta)T_\xi + (x_\zeta y_\xi - x_\xi y_\zeta)T_\eta + (x_\xi y_\eta - x_\eta y_\xi)T_\zeta][x_\eta y_\zeta - x_\zeta y_\eta]) \\
&/(J\sqrt{(y_\eta z_\zeta - y_\zeta z_\eta)^2 + (x_\eta z_\zeta - x_\zeta z_\eta)^2 + (x_\eta y_\zeta - x_\zeta y_\eta)^2}) \\
&= h_3(T_{M,j,k} - T_{\infty_3})
\end{aligned} \tag{4.68}$$

By substituting the relations (4.62) to (4.64) into Equation (4.68) and then solving for $T_{M,j,k}$, a relationship is derived which allows us to obtain the temperature distribution on the surface Γ_3. Similar relationships may be

derived for other boundaries with a Robin condition. Furthermore, if radiation heat transfer is present on a given surface, we can simply write the radiation equation on the right hand side of Equation (4.68) and then solve it for $T_{M,j,k}$ using the Newton-Raphson method. In case of boundary conditions involving a prescribed heat flux, we can simply write the heat flux term instead of the convective heat transfer term.

4.3 Field solver validation

To validate the implementation of the numerical scheme adopted so far, the temperature distribution for two heat conducting bodies are compared to the temperature distribution obtained from the commercial software ANSYS. The first benchmark case is illustrated in Figure 4-5 (**Benchmark Problem 1**). The semicircular surface $BEHC$ is subject to a prescribed heat flux $\dot{q}$ and the other faces are exposed to the convective heat transfer. The numerical values for the coefficients involved in the problem and the coordinates of the physical domain are listed in Table 4-1.

$\dot{q}_{\text{on face BEHC}} = 200 \frac{\text{W}}{\text{m}^2}$	$h_{\text{other faces}} = 3 \frac{\text{W}}{\text{m}^2.\text{C}}$
$T_{\infty(\text{other faces})} = 30^{\circ}\text{C}$	$k_T = 15 \frac{\text{W}}{\text{m.C}}$
$B(1,1,1) = (5,1,1)\text{m}$	$E(M,1,1) = (15,1,1)\text{m}$
$H(M,1,L) = (15,1,21)\text{m}$	$C(1,1,L) = (5,1,21)\text{m}$
$A(1,N,1) = (-5,1,1)\text{m}$	$F(M,N,1) = (25,1,1)\text{m}$
$G(M,N,L) = (25,1,21)\text{m}$	$D(1,N,L) = (-5,1,21)\text{m}$

Table 4-1 Parameters used to validate the numerical scheme by ANSYS (**Benchmark Problem 1**).

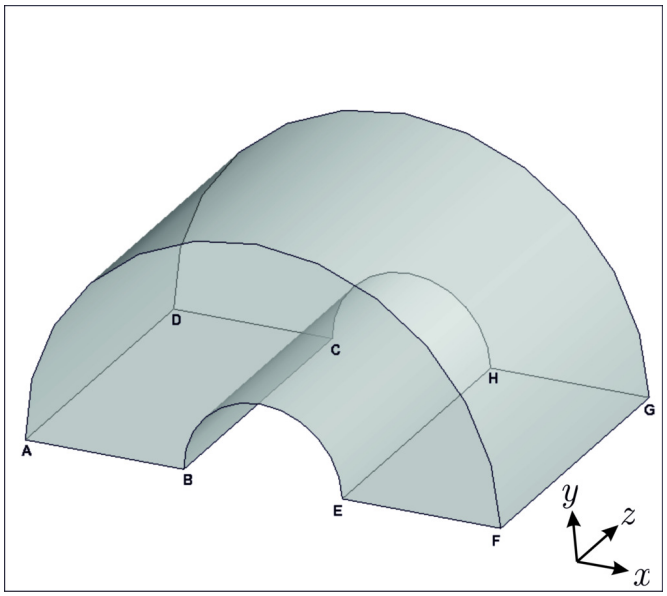


Figure 4-5 The heat conducting body subject to the prescribed boundary conditions (**Benchmark Problem 1**).

Employing the numerical procedure explained above with the boundary conditions listed in Table 4-1, the temperature distribution shown in Figure 4-6 is obtained. In order to solve the equations, the Successive Over Relaxation is used. The tolerance used in the iterative process is 10^{-7} and variables are double precision.

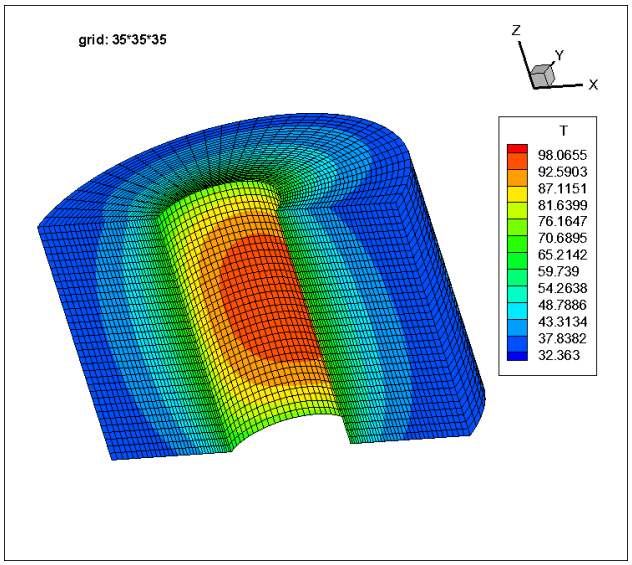


Figure 4-6 The grid and temperature contours of the heated body specified by the data listed in Table 4-1.

By employing the finite element software ANSYS (V. 12) for approximately the same grid size, the temperature distribution shown in Figure 4-7 is obtained.

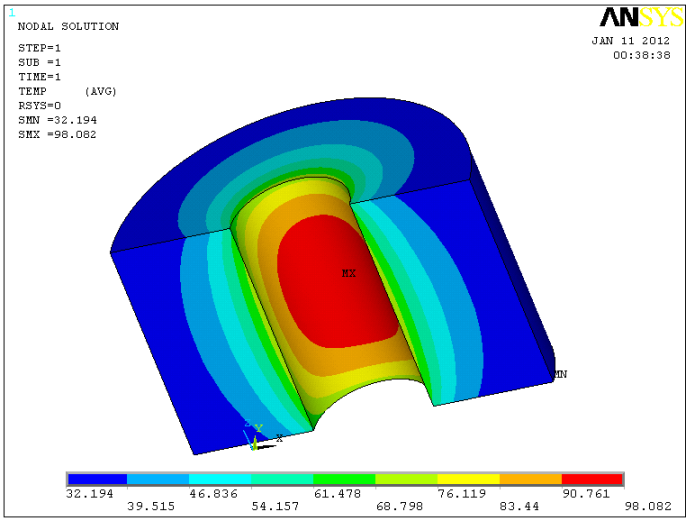


Figure 4-7 Temperature contours obtained by ANSYS (**Benchmark Problem 1**).

The percent difference of the minimum and maximum temperatures resulting from the procedure based on body-fitted grid generation and ANSYS is obtained as follows:

$$\text{Diff}_{max} = \left| \frac{(T_{max})_{\text{BFGG}} - (T_{max})_{\text{ANSYS}}}{(\frac{(T_{max})_{\text{BFGG}} + (T_{max})_{\text{ANSYS}}}{2})} \right| \times 100 = 0.0168\%$$

$$\text{Diff}_{min} = \left| \frac{(T_{min})_{\text{BFGG}} - (T_{min})_{\text{ANSYS}}}{(\frac{(T_{min})_{\text{BFGG}} + (T_{min})_{\text{ANSYS}}}{2})} \right| \times 100 = 0.52\%$$

The second validation case (**Benchmark Problem 2**) is the cube shown in Figure 4-8. The surface $BEHC$ is subject to a prescribed heat flux $\dot{q}$ and the other faces are exposed to the convective heat transfer. The numerical values for the coefficients involved in the problem and the coordinates of the physical domain are listed in Table 4-2.

$\dot{q}_{\text{on face BEHC}} = 700 \frac{\text{W}}{\text{m}^2}$	$h_{\text{other faces}} = 3 \frac{\text{W}}{\text{m}^2.\text{C}}$
$T_{\infty(\text{other faces})} = 40^{\circ}\text{C}$	$k_T = 60 \frac{\text{W}}{\text{m.C}}$
$B(1,1,1) = (10,10,10)\text{m}$	$E(M,1,1) = (50,10,10)\text{m}$
$H(M,1,L) = (50,10,60)\text{m}$	$C(1,1,L) = (10,10,60)\text{m}$
$A(1,N,1) = (10,70,10)\text{m}$	$F(M,N,1) = (50,70,10)\text{m}$
$G(M,N,L) = (50,70,60)\text{m}$	$D(1,N,L) = (10,70,60)\text{m}$

Table 4-2 Data used to validate the numerical implementation by ANSYS (**Benchmark Problem 2**).

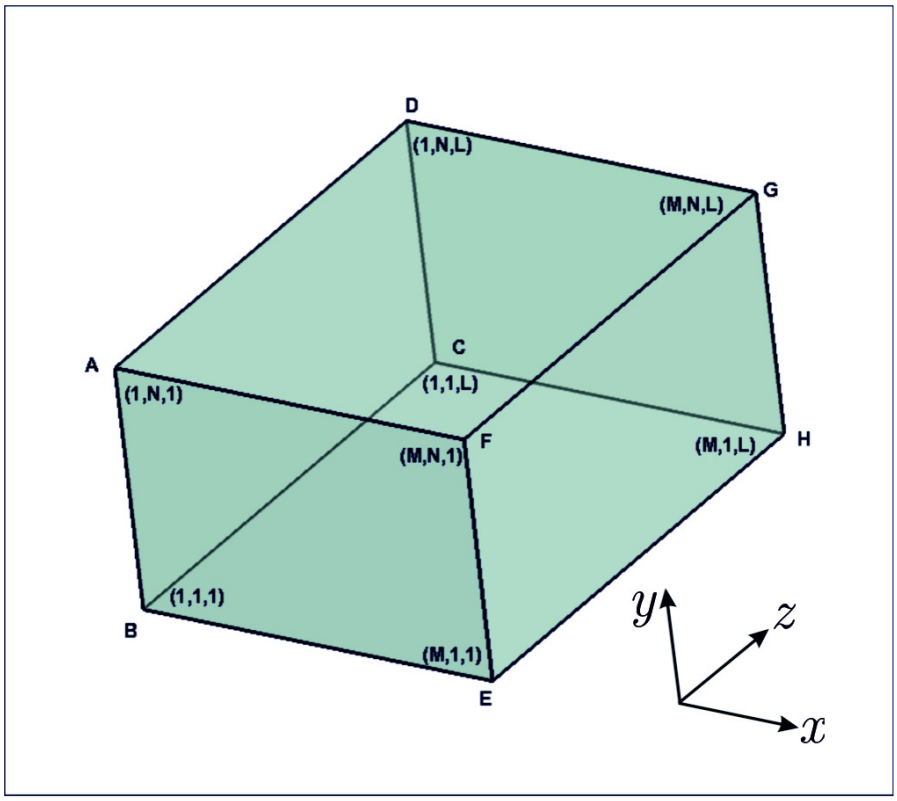


Figure 4-8 The heat conducting body (cube) subject to the prescribed boundary conditions (**Benchmark Problem 2**).

Similarly, employing the numerical procedure explained above with the boundary conditions listed in Table 4-2 and a $30 \times 30 \times 30$ mesh, the temperature distribution in the physical domain can be obtained and is illustrated in Fig 9.

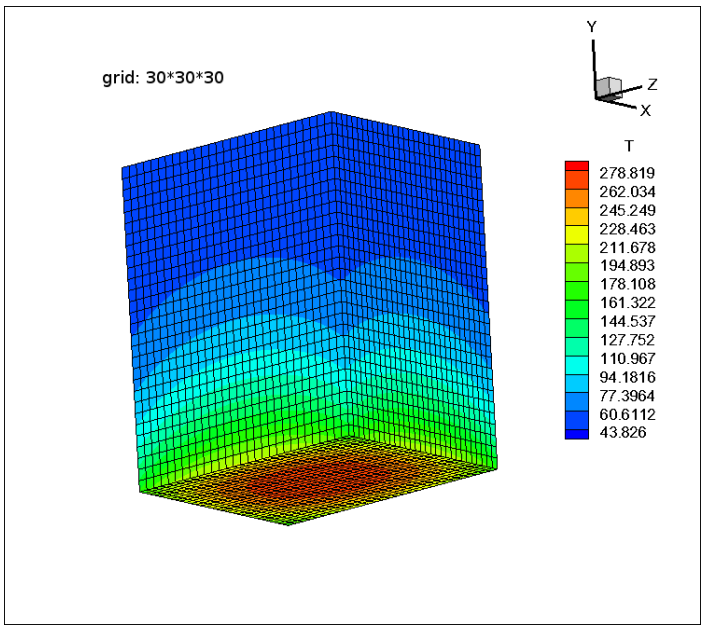


Figure 4-9 Grid and temperature contours of the heated body specified by the data listed in Table 2 (**Benchmark Problem 2**).

By using the ANSYS for approximately the same grid size, we obtain the temperature distribution shown in Figure 4-10.

Using the definitions introduced above, the difference in maximum temperatures is %0.231 and the difference in minimum temperatures is %0.588 .

The above percent differences reveal a good agreement between the results obtained from the use of the FDM (employing the body-fitted grid generation technique) and FEM (using ANSYS). Thus, it confirms the correct implementation of the proposed algorithm for the direct solver.

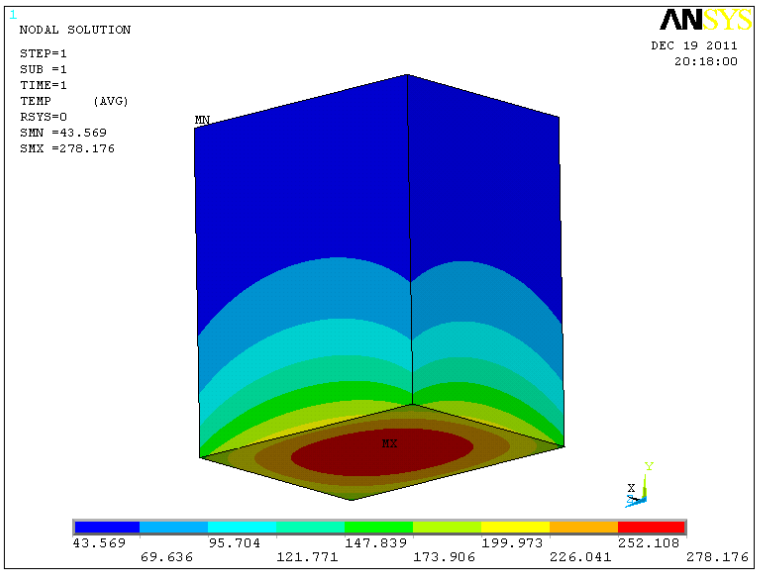


Figure 4-10 The temperature contours obtained by ANSYS (**Benchmark Problem 2**).

The next chapter deals with the 3-D shape optimization in heat transfer problems and the involved sensitivity analysis. The 3-D heat transfer problem and the associated objective function will be defined; the sensitivity analysis will be developed; and finally the conjugate gradient method will be employed to infer the optimal shape design.

5. Three dimensional sensitivity analysis and shape optimization

5.1 Introduction

The mathematical representation of the heat conduction problem in the shape optimization problem of interest here (see Figure 4-3) can be expressed as

$$\nabla^2 T = 0 \quad \text{in physical domain } \Omega \tag{5.1}$$

subject to the boundary conditions

$$\frac{\partial T}{\partial n} = -\frac{\dot{q}}{k_T} \quad \text{on boundary surface } \Gamma_5 \tag{5.2}$$

$$\frac{\partial T}{\partial n} = -\frac{h}{k_T}(T_{\Gamma_i} - T_{\infty_i}) \quad \text{on boundary surface } \Gamma_i, i = 1,2,3,4,6 \tag{5.3}$$

The aim of the shape optimization problem considered in this study is to find the optimal shape for the outer surface Γ_6 in Figure 4-3 so that the square of the difference between the outer surface temperature and a desired outer temperature distribution is minimized. In more formal mathematical terms, the shape optimization problem can be expressed as

$$\min_{(x,y,z)\in\Gamma_6} \left\{ \mathcal{J}(x,y,z) := \| T_{(x,y,z)} - T_d \|^2 : \nabla^2 T = 0 \text{ in } \Omega, \text{BCs in (5.2) \& (5.3)} \right\} \tag{5.4}$$

where x, y, and z are coordinates of the nodes at the boundary. The "design parameters" x, y, and z define the shape of the outer surface Γ_6. Let $x^{(k)}_{i,N,k}$, $y^{(k)}_{i,N,k}$, and $z^{(k)}_{i,N,k}$ $(i = 1,2,3,\ldots,M; k = 1,2,3,\ldots,L)$ be the

coordinates of node $F_{i,N,k}$ on the outer boundary to be optimized at iteration k (see Figure 5-1). The location of the nodes on this outer surface has to be numerically optimized for the objective function

$$\mathcal{J} = \sum_{k=1}^{L}\sum_{i=1}^{M}(T_{i,N,k} - T_d)^2 \tag{5.5}$$

to reach a minimum. $T_{i,N,k}$ and T_d are the nodal temperature at point $F_{i,N,k}$ $(i = 1,2,3,\ldots,M; k = 1,2,3,\ldots,L)$ and the desired outer surface temperature, respectively. Because the shape of the body of interest at iteration k is defined by the locations of the nodes on the outer surface, the objective function is dependent on the locations of the nodes.

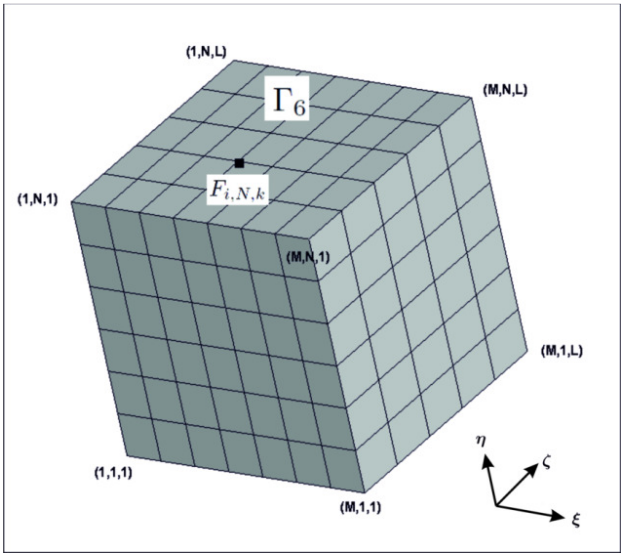


Figure 5-1 Illustration of the computational domain, associated mesh, and notations. The location of the nodes on the outer upper surface Γ_6 has to be numerically optimized for the objective function value to reach a minimum

5.2 Sensitivity analysis

Suppose we wish to calculate the sensitivity of temperature of nodes on the boundary Γ_6 (see Figure 5-1), $T_{i,N,k}\,(i=1,2,3,\ldots,M\,;k=1,2,3,\ldots,L)$, to the $x-$position of the nodes on the boundary Γ_6, $x_{i',N,k'}\,(i'=1,2,3,\ldots,M\,;\ k'=1,2,3,\ldots,L)$. The sensitivity analysis can be performed by introducing small perturbations to the $x-$coordinate of each point on the boundary, Γ_6, $x_{i',N,k'}$, individually. The grid generation and direct problem may be solved for this perturbated shape to obtain the new temperatures $T_{i,N,k}$. Using these values for the temperatures, one can evaluate the dependency of the temperatures $T_{i,N,k}$ to the perturbation of the $x-$position of points of coordinates (i',N,k'). The finite difference method may be used to formulate these perturbations as follows

$$\frac{\partial T_{i,N,k}}{\partial x_{i',N,k'}}=\frac{T_{i,N,k}(x_{i',N,k'}+\varepsilon x_{i',N,k'})-T_{i,N,k}(x_{i',N,k'})}{\varepsilon x_{i',N,k'}}$$

where ε may be, say, 10^{-6}. The term $\varepsilon x_{i',N,k'}$ is the perturbation in the $x-$position of points of coordinates (i',N,k'). Since the sensitivity of each temperature $T_{i,N,k}\,(i=1,2,3,\ldots,M\,;k=1,2,3,\ldots,L)$ to each $x-$position of points of coordinates $(i',N,k')\,(i'=1,2,3,\ldots,M\,;k'=1,2,3,\ldots,L)$ is required, the computation of the sensitivity coefficients using this method requires $(M\times L)$ additional solution of direct heat transfer problem. In other words, by introducing a small perturbation to the $x-$coordinate of each point on the boundary, one can evaluate the dependency of all temperatures on the boundary to that perturbated coordinate. This forms every column of the sensitivity matrix. ML is devoted to the number of nodes on Γ_6 for considering the $x-$position of points of coordinates (i',N,k'). Hence, this method is suitable when the number of points on the boundary surface Γ_6, $M\times L$, is small. For example, if a grid of size $M\times N\times L=\ 15\times15\times15$ is used to mesh the 3-D body, the perturbation method demands $M\times L=15\times15=225$ additional solution of direct solver at any iteration. For a shape optimization problem of such a body with, say, 100 iterations, the shape optimization process requires $100\times225=22500$ additional solution of direct

solver. Therefore the computational cost will be very high which makes the use of the perturbation method in 3-D heat transfer problems impractical. In addition, introducing the y and z coordinates makes the process more complex. As mentioned before, we will introduce a novel method to calculate the sensitivity coefficients based on the finite difference method and transformation rules introduced in Field Equation Solver Section (Section 4.2). In this chapter, the relationships derived previously for 2-D state will be expanded to 3-D geometry. In the following, the sensitivity matrix (Jacobian matrix) for three dimensional shape optimization problems is introduced and the method for the calculation of the sensitivity coefficients is discussed. In this study, it is aimed to evaluate the sensitivity of the objective function defined by Equation (5.5) to the location of nodes on the surface Γ_6 in Figure 5-1.

In order to calculate the sensitivity coefficients, the following equations may be written

$$\frac{\partial \boldsymbol{\mathcal{J}}}{\partial x_{i',N,k'}} = 2\sum_{k=1}^{L}\sum_{i=1}^{M}(T_{i,N,k} - T_d)\frac{\partial T_{i,N,k}}{\partial x_{i',N,k'}} \tag{5.6}$$

$$\frac{\partial \boldsymbol{\mathcal{J}}}{\partial y_{i',N,k'}} = 2\sum_{k=1}^{L}\sum_{i=1}^{M}(T_{i,N,k} - T_d)\frac{\partial T_{i,N,k}}{\partial y_{i',N,k'}} \tag{5.7}$$

$$\frac{\partial \boldsymbol{\mathcal{J}}}{\partial z_{i',N,k'}} = 2\sum_{k=1}^{L}\sum_{i=1}^{M}(T_{i,N,k} - T_d)\frac{\partial T_{i,N,k}}{\partial z_{i',N,k'}} \tag{5.8}$$

where $i' \in [1, M]$, $k' \in [1, L]$. The expressions $\dfrac{\partial T_{i,N,k}}{\partial x_{i',N,k'}}$, $\dfrac{\partial T_{i,N,k}}{\partial y_{i',N,k'}}$, and $\dfrac{\partial T_{i,N,k}}{\partial z_{i',N,k'}}$ in the above relations are called the Jacobian coefficients. The associated sensitivity matrices, therefore, can be explicitly written as

$$
\mathbf{Ja}_x = \left\{ \begin{array}{ccccc}
\overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,1}}{\partial x_{1,N,1}} & \frac{\partial T_{1,N,1}}{\partial x_{2,N,1}} & \cdots \\ \frac{\partial T_{2,N,1}}{\partial x_{1,N,1}} & \frac{\partial T_{2,N,1}}{\partial x_{2,N,1}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,1}}{\partial x_{1,N,1}} & \frac{\partial T_{M,N,1}}{\partial x_{2,N,1}} & \cdots \end{array}}^{k'=1} & & \frac{\partial T_{M,N,1}}{\partial x_{M,N,1}} & \cdots & \overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,1}}{\partial x_{1,N,L}} & \frac{\partial T_{1,N,1}}{\partial x_{2,N,L}} & \cdots \\ \frac{\partial T_{2,N,1}}{\partial x_{1,N,L}} & \frac{\partial T_{2,N,1}}{\partial x_{2,N,L}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,1}}{\partial x_{1,N,L}} & \frac{\partial T_{M,N,1}}{\partial x_{2,N,L}} & \cdots \end{array}}^{k'=L} \\
\overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,2}}{\partial x_{1,N,1}} & \frac{\partial T_{1,N,2}}{\partial x_{2,N,1}} & \cdots \\ \frac{\partial T_{2,N,2}}{\partial x_{1,N,1}} & \frac{\partial T_{2,N,2}}{\partial x_{2,N,1}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,2}}{\partial x_{1,N,1}} & \frac{\partial T_{M,N,2}}{\partial x_{2,N,1}} & \cdots \end{array}}^{k'=1} & & \frac{\partial T_{M,N,2}}{\partial x_{M,N,1}} & \cdots & \overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,2}}{\partial x_{1,N,L}} & \frac{\partial T_{1,N,2}}{\partial x_{2,N,L}} & \cdots \\ \frac{\partial T_{2,N,2}}{\partial x_{1,N,L}} & \frac{\partial T_{2,N,2}}{\partial x_{2,N,L}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,2}}{\partial x_{1,N,L}} & \frac{\partial T_{M,N,2}}{\partial x_{2,N,L}} & \cdots \end{array}}^{k'=L} \\
\vdots & & \ddots & & \vdots \\
\overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,L}}{\partial x_{1,N,1}} & \frac{\partial T_{1,N,L}}{\partial x_{2,N,1}} & \cdots \\ \frac{\partial T_{2,N,L}}{\partial x_{1,N,1}} & \frac{\partial T_{2,N,L}}{\partial x_{2,N,1}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,L}}{\partial x_{1,N,1}} & \frac{\partial T_{M,N,L}}{\partial x_{2,N,1}} & \cdots \end{array}}^{k'=1} & & \frac{\partial T_{M,N,L}}{\partial x_{M,N,1}} & \cdots & \overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,L}}{\partial x_{1,N,L}} & \frac{\partial T_{1,N,L}}{\partial x_{2,N,L}} & \cdots \\ \frac{\partial T_{2,N,L}}{\partial x_{1,N,L}} & \frac{\partial T_{2,N,L}}{\partial x_{2,N,L}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,L}}{\partial x_{1,N,L}} & \frac{\partial T_{M,N,L}}{\partial x_{2,N,L}} & \cdots \end{array}}^{k'=L}
\end{array} \right\} \tag{5.9}
$$

$$
\mathbf{Ja}_y = \left\{ \begin{array}{ccccc}
\overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,1}}{\partial y_{1,N,1}} & \frac{\partial T_{1,N,1}}{\partial y_{2,N,1}} & \cdots \\ \frac{\partial T_{2,N,1}}{\partial y_{1,N,1}} & \frac{\partial T_{2,N,1}}{\partial y_{2,N,1}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,1}}{\partial y_{1,N,1}} & \frac{\partial T_{M,N,1}}{\partial y_{2,N,1}} & \cdots \end{array}}^{k'=1} & & \frac{\partial T_{1,N,1}}{\partial y_{M,N,1}} & \cdots & \overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,1}}{\partial y_{1,N,L}} & \frac{\partial T_{1,N,1}}{\partial y_{2,N,L}} & \cdots \\ \frac{\partial T_{2,N,1}}{\partial y_{1,N,L}} & \frac{\partial T_{2,N,1}}{\partial y_{2,N,L}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,1}}{\partial y_{1,N,L}} & \frac{\partial T_{M,N,1}}{\partial y_{2,N,L}} & \cdots \end{array}}^{k'=L} & \frac{\partial T_{1,N,1}}{\partial y_{M,N,L}} \\
\overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,2}}{\partial y_{1,N,1}} & \frac{\partial T_{1,N,2}}{\partial y_{2,N,1}} & \cdots \\ \frac{\partial T_{2,N,2}}{\partial y_{1,N,1}} & \frac{\partial T_{2,N,2}}{\partial y_{2,N,1}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,2}}{\partial y_{1,N,1}} & \frac{\partial T_{M,N,2}}{\partial y_{2,N,1}} & \cdots \end{array}}^{k'=1} & & \frac{\partial T_{1,N,2}}{\partial y_{M,N,1}} & \cdots & \overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,2}}{\partial y_{1,N,L}} & \frac{\partial T_{1,N,2}}{\partial y_{2,N,L}} & \cdots \\ \frac{\partial T_{2,N,2}}{\partial y_{1,N,L}} & \frac{\partial T_{2,N,2}}{\partial y_{2,N,L}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,2}}{\partial y_{1,N,L}} & \frac{\partial T_{M,N,2}}{\partial y_{2,N,L}} & \cdots \end{array}}^{k'=L} & \frac{\partial T_{1,N,2}}{\partial y_{M,N,L}} \\
\vdots & & \vdots & & \vdots \\
\overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,L}}{\partial y_{1,N,1}} & \frac{\partial T_{1,N,L}}{\partial y_{2,N,1}} & \cdots \\ \frac{\partial T_{2,N,L}}{\partial y_{1,N,1}} & \frac{\partial T_{2,N,L}}{\partial y_{2,N,1}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,L}}{\partial y_{1,N,1}} & \frac{\partial T_{M,N,L}}{\partial y_{2,N,1}} & \cdots \end{array}}^{k'=1} & & \frac{\partial T_{1,N,L}}{\partial y_{M,N,1}} & \cdots & \overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,L}}{\partial y_{1,N,L}} & \frac{\partial T_{1,N,L}}{\partial y_{2,N,L}} & \cdots \\ \frac{\partial T_{2,N,L}}{\partial y_{1,N,L}} & \frac{\partial T_{2,N,L}}{\partial y_{2,N,L}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,L}}{\partial y_{1,N,L}} & \frac{\partial T_{M,N,L}}{\partial y_{2,N,L}} & \cdots \end{array}}^{k'=L} & \frac{\partial T_{1,N,L}}{\partial y_{M,N,L}}
\end{array} \right\}
$$

(5.10)

$$
\mathbf{Ja}_z = \left\{ \begin{array}{ccccc}
\overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,1}}{\partial z_{1,N,1}} & \frac{\partial T_{1,N,1}}{\partial z_{2,N,1}} & \cdots \\ \frac{\partial T_{2,N,1}}{\partial z_{1,N,1}} & \frac{\partial T_{2,N,1}}{\partial z_{2,N,1}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,1}}{\partial z_{1,N,1}} & \frac{\partial T_{M,N,1}}{\partial z_{2,N,1}} & \cdots \end{array}}^{k'=1} & & \cdots & & \overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,1}}{\partial z_{1,N,L}} & \frac{\partial T_{1,N,1}}{\partial z_{2,N,L}} & \cdots \\ \frac{\partial T_{2,N,1}}{\partial z_{1,N,L}} & \frac{\partial T_{2,N,1}}{\partial z_{2,N,L}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,1}}{\partial z_{1,N,L}} & \frac{\partial T_{M,N,1}}{\partial z_{2,N,L}} & \cdots \end{array}}^{k'=L} \\
\overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,2}}{\partial z_{1,N,1}} & \frac{\partial T_{1,N,2}}{\partial z_{2,N,1}} & \cdots \\ \frac{\partial T_{2,N,2}}{\partial z_{1,N,1}} & \frac{\partial T_{2,N,2}}{\partial z_{2,N,1}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,2}}{\partial z_{1,N,1}} & \frac{\partial T_{M,N,2}}{\partial z_{2,N,1}} & \cdots \end{array}}^{k'=1} & & \cdots & & \overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,2}}{\partial z_{1,N,L}} & \frac{\partial T_{1,N,2}}{\partial z_{2,N,L}} & \cdots \\ \frac{\partial T_{2,N,2}}{\partial z_{1,N,L}} & \frac{\partial T_{2,N,2}}{\partial z_{2,N,L}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,2}}{\partial z_{1,N,L}} & \frac{\partial T_{M,N,2}}{\partial z_{2,N,L}} & \cdots \end{array}}^{k'=L} \\
\vdots & & \ddots & & \vdots \\
\overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,L}}{\partial z_{1,N,1}} & \frac{\partial T_{1,N,L}}{\partial z_{2,N,1}} & \cdots \\ \frac{\partial T_{2,N,L}}{\partial z_{1,N,1}} & \frac{\partial T_{2,N,L}}{\partial z_{2,N,1}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,L}}{\partial z_{1,N,1}} & \frac{\partial T_{M,N,L}}{\partial z_{2,N,1}} & \cdots \end{array}}^{k'=1} & & \cdots & & \overbrace{\begin{array}{ccc} \frac{\partial T_{1,N,L}}{\partial z_{1,N,L}} & \frac{\partial T_{1,N,L}}{\partial z_{2,N,L}} & \cdots \\ \frac{\partial T_{2,N,L}}{\partial z_{1,N,L}} & \frac{\partial T_{2,N,L}}{\partial z_{2,N,L}} & \cdots \\ \vdots & & \\ \frac{\partial T_{M,N,L}}{\partial z_{1,N,L}} & \frac{\partial T_{M,N,L}}{\partial z_{2,N,L}} & \cdots \end{array}}^{k'=L}
\end{array} \right\}
$$

(5.11)

The rank of the sensitivity matrix is $M \times L$ by $M \times L$. Therefore, the computation of the sensitivity matrix for a finely meshed outer surface demands high storage and is computationally intensive. For example, for a grid of size $40 \times 40 \times 40$, the Jacobian matrix will be of rank 1600×1600.

Since the physical domain is mapped onto the computational one, the chain rule may be used to correlate the variables in the two domains. Therefore,

$$\frac{\partial T_{i,N,k}}{\partial x_{i',N,k'}} = \frac{\partial T_{i,N,k}}{\partial \xi}\frac{\partial \xi}{\partial x_{i',N,k'}} + \frac{\partial T_{i,N,k}}{\partial \eta}\frac{\partial \eta}{\partial x_{i',N,k'}} + \frac{\partial T_{i,N,k}}{\partial \zeta}\frac{\partial \zeta}{\partial x_{i',N,k'}} \tag{5.12}$$

$$\frac{\partial T_{i,N,k}}{\partial y_{i',N,k'}} = \frac{\partial T_{i,N,k}}{\partial \xi}\frac{\partial \xi}{\partial y_{i',N,k'}} + \frac{\partial T_{i,N,k}}{\partial \eta}\frac{\partial \eta}{\partial y_{i',N,k'}} + \frac{\partial T_{i,N,k}}{\partial \zeta}\frac{\partial \zeta}{\partial y_{i',N,k'}} \tag{5.13}$$

$$\frac{\partial T_{i,N,k}}{\partial z_{i',N,k'}} = \frac{\partial T_{i,N,k}}{\partial \xi}\frac{\partial \xi}{\partial z_{i',N,k'}} + \frac{\partial T_{i,N,k}}{\partial \eta}\frac{\partial \eta}{\partial z_{i',N,k'}} + \frac{\partial T_{i,N,k}}{\partial \zeta}\frac{\partial \zeta}{\partial z_{i',N,k'}} \tag{5.14}$$

By interchanging x and ξ, y and η, and z and ζ, and solving the derived equations for $\frac{\partial T}{\partial x}$, $\frac{\partial T}{\partial y}$, and $\frac{\partial T}{\partial z}$, we obtain

$$\begin{aligned}\frac{\partial T_{i,N,k}}{\partial x_{i',N,k'}} = [&(y_\xi)_{i',N,k'}(z_\eta)_{i',N,k'}(T_\zeta)_{i,N,k} - (y_\xi)_{i',N,k'}(z_\zeta)_{i',N,k'}(T_\eta)_{i,N,k} \\ &+(y_\zeta)_{i',N,k'}(z_\xi)_{i',N,k'}(T_\eta)_{i,N,k} - (T_\zeta)_{i,N,k}(z_\xi)_{i',N,k'}(y_\eta)_{i',N,k'} \\ &+(T_\xi)_{i,N,k}(z_\zeta)_{i',N,k'}(y_\eta)_{i',N,k'} - (T_\xi)_{i,N,k}(y_\zeta)_{i',N,k'}(z_\eta)_{i',N,k'}]\,/\,J\end{aligned} \tag{5.15}$$

$$\begin{aligned}\frac{\partial T_{i,N,k}}{\partial y_{i',N,k'}} = -[&(x_\xi)_{i',N,k'}(z_\eta)_{i',N,k'}(T_\zeta)_{i,N,k} - (x_\xi)_{i',N,k'}(z_\zeta)_{i',N,k'}(T_\eta)_{i,N,k} \\ &+(x_\zeta)_{i',N,k'}(z_\xi)_{i',N,k'}(T_\eta)_{i,N,k} - (T_\zeta)_{i,N,k}(z_\xi)_{i',N,k'}(x_\eta)_{i',N,k'} \\ &+(T_\xi)_{i,N,k}(z_\zeta)_{i',N,k'}(x_\eta)_{i',N,k'} - (T_\xi)_{i,N,k}(x_\zeta)_{i',N,k'}(z_\eta)_{i',N,k'}]\,/\,J\end{aligned} \tag{5.16}$$

$$\begin{aligned}\frac{\partial T_{i,N,k}}{\partial z_{i',N,k'}} = [&(x_\xi)_{i',N,k'}(y_\eta)_{i',N,k'}(T_\zeta)_{i,N,k} - (x_\xi)_{i',N,k'}(y_\zeta)_{i',N,k'}(T_\eta)_{i,N,k} \\ &+(x_\zeta)_{i',N,k'}(y_\xi)_{i',N,k'}(T_\eta)_{i,N,k} - (T_\zeta)_{i,N,k}(y_\xi)_{i',N,k'}(x_\eta)_{i',N,k'} \\ &+(T_\xi)_{i,N,k}(y_\zeta)_{i',N,k'}(x_\eta)_{i',N,k'} - (T_\xi)_{i,N,k}(x_\zeta)_{i',N,k'}(y_\eta)_{i',N,k'}]\,/\,J\end{aligned} \tag{5.17}$$

where J is the Jacobian of the transformation [see Equation (4.10)]. Now using the finite difference method to discretize the equations in the computational domain, we can write appropriate algebraic approximations for all partial derivatives involved in the above equations. For example,

$$(T_\xi)_{i,N,k} = \frac{1}{2}(T_{i+1,N,k} - T_{i-1,N,k}) \tag{5.18}$$

$$(T_\eta)_{i,N,k} = \frac{1}{2}(3T_{i,N,k} - 4T_{i,N-1,k} + T_{i,N-2,k}) \tag{5.19}$$

$$(T_\zeta)_{i,N,k} = \frac{1}{2}(T_{i,N,k+1} - T_{i,N,k-1}) \tag{5.20}$$

where $i \in [2, M-1]$ and $k \in [2, L-1]$. The same procedure can be followed to find the relations for $(x_\xi)_{i',N,k'}$, $(x_\eta)_{i',N,k'}$, $(x_\zeta)_{i',N,k'}$, $(y_\xi)_{i',N,k'}$, $(y_\eta)_{i',N,k'}$, $(y_\zeta)_{i',N,k'}$, $(z_\xi)_{i',N,k'}$, $(z_\eta)_{i',N,k'}$, and $(z_\zeta)_{i',N,k'}$ where $i' \in [2, M-1]$ and $k' \in [2, L-1]$. Furthermore, for $i = 1$, $i = M$, $k = 1$, $k = L$, $i' = 1$, $i' = M$, $k' = 1$, and $k' = L$ one can use the one-sided forward and one-sided backward relations.

5.3 The Conjugate Gradient Method

The CGM-based algorithm for the 2-D heat transfer problem was discussed in Section 3.7. It can be easily expanded to the 3-D heat transfer problem.

5.4 Stopping criteria

The stopping criteria used for the 2-D problems [Equations (3.25) and (3.26)] can also be employed for the 3-D problems.

5.5 Results and discussion

Before proceeding to the shape optimization using the CGM, it is necessary to fully specify the problem and the associated boundary conditions. For the first and second test cases, consider the physical domain shown in Figure 5-2.

The surface $BEHC$ is subject to a prescribed heat flux $\dot{q}$ and the other faces are exposed to the convective heat transfer.

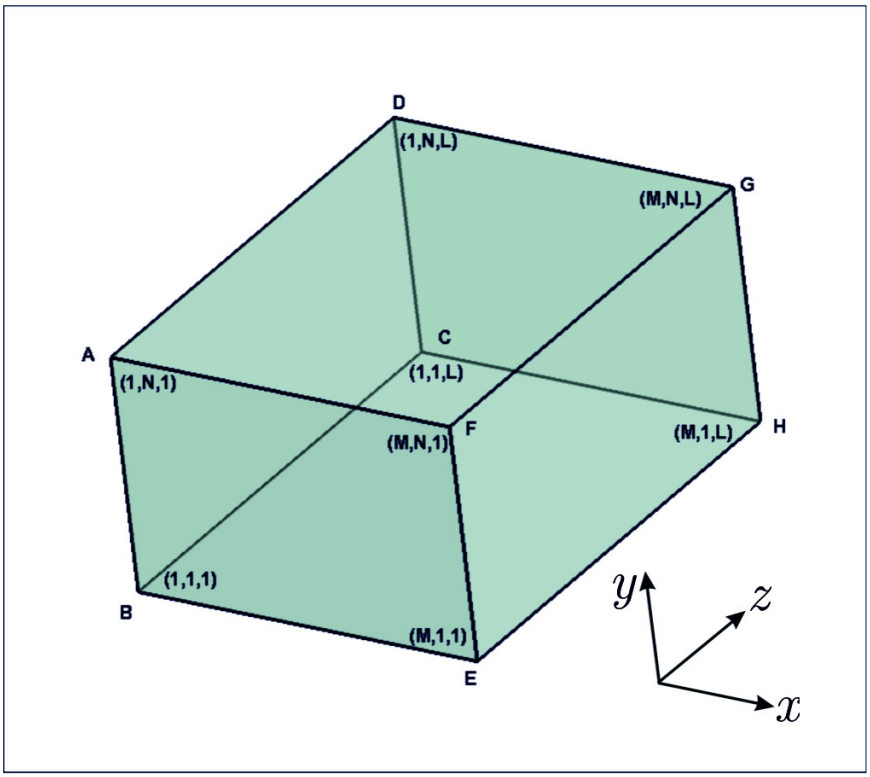


Figure 5-2 The heat conducting body subject to the prescribed boundary conditions.

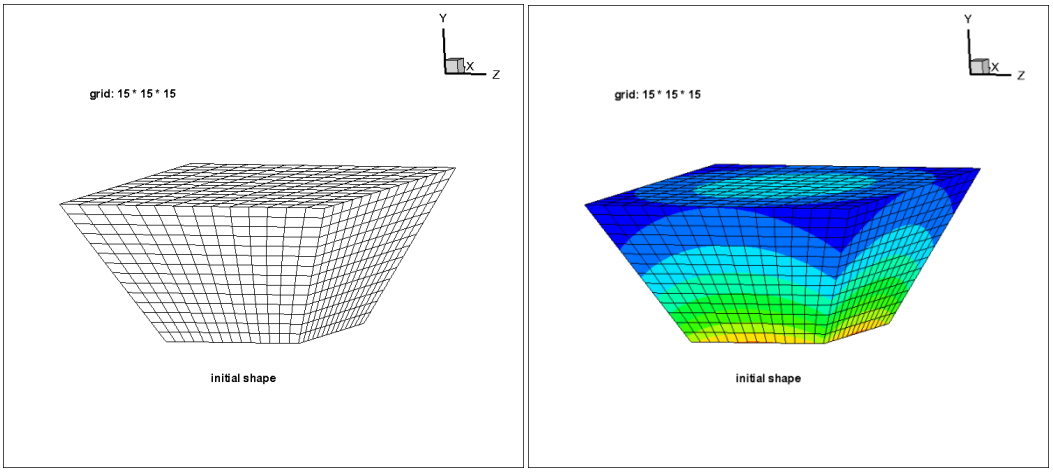
Test case 1

The geometry considered is shown in Figure 5-2 and the numerical values of the coefficients involved in the problem and the coordinates of the physical domain are listed in Table 5-1.

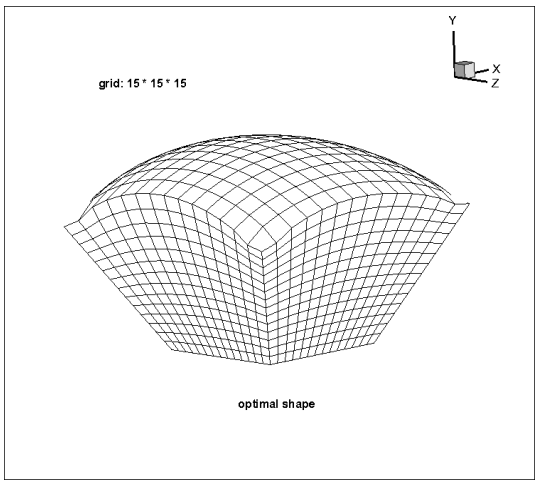
$\dot{q}_{\text{on face BEHC}} = 1000\,\frac{\text{W}}{\text{m}^2}$	$h_{\text{other faces}} = 4\,\frac{\text{W}}{\text{m}^2.\text{C}}$
$T_{\infty(\text{other faces})} = 30^{\circ}\text{C}$	$k_T = 55\,\frac{\text{W}}{\text{m.C}}$
$B(1,1,1) = (1,1,1)\text{m}$	$E(M,1,1) = (5,1,1)\text{m}$
$H(M,1,L) = (5,1,5)\text{m}$	$C(1,1,L) = (1,1,5)\text{m}$
$A(1,N,1) = (-1,5,-1)\text{m}$	$F(M,N,1) = (7,5,-1)\text{m}$
$G(M,N,L) = (7,5,7)\text{m}$	$D(1,N,L) = (-1,5,7)\text{m}$
$T_d = 50^{\circ}\text{C}$	mesh size=$15\times15\times15$

Table 5-1 Data used for Test case 1

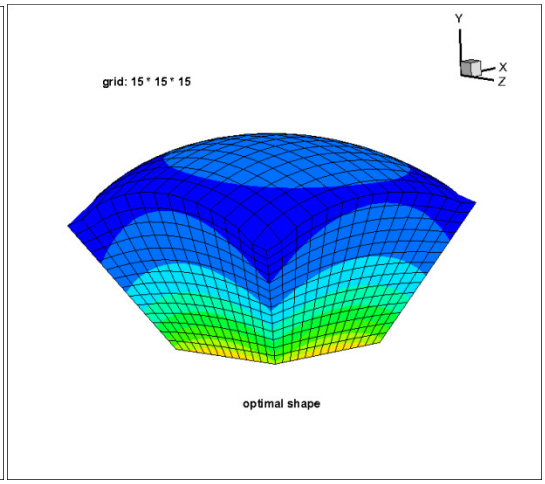
The aim of the optimization is to find the optimal shape featuring a uniform temperature on the outer surface $(AFGD)$ for the prescribed boundary conditions given in Table 5-1. The results for this first Test case are depicted in Figure 5-3.



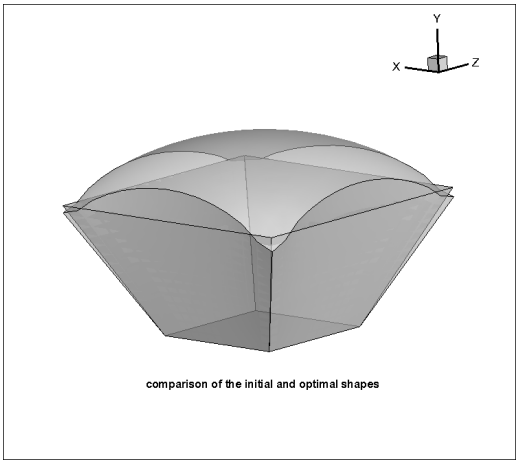
a) b)


Y
X
Z
grid: 15 * 15 * 15
optimal shape

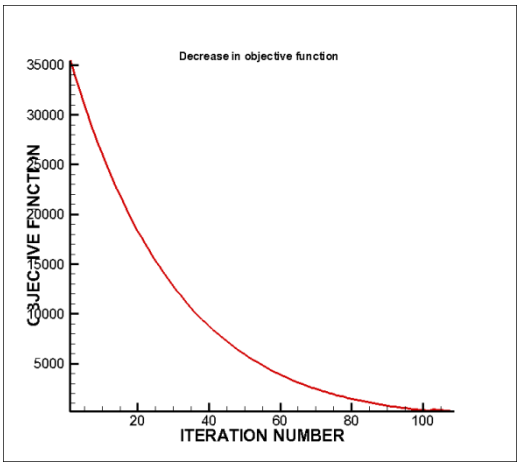
c)


Y
X
Z
grid: 15 * 15 * 15
optimal shape

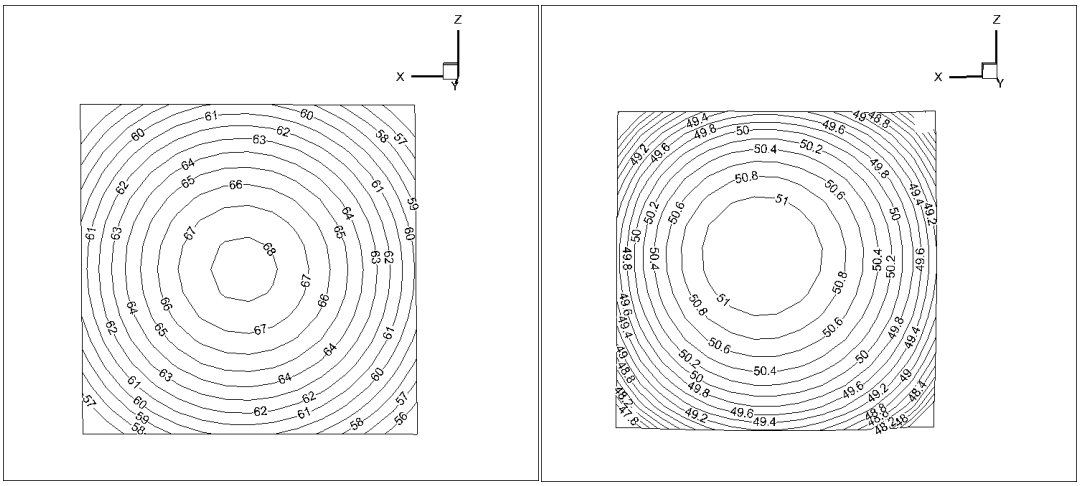
d)


Y
X
Z
comparison of the initial and optimal shapes

e)


Decrease in objective function
35000
30000
25000
20000
15000
10000
5000
OBJECTIVE FUNCTION
20
40
60
80
100
ITERATION NUMBER

f)



g) h)

Figure 5-3 a) meshed initial shape; b) meshed initial shape and its temperature contours; c) meshed optimal shape; d) meshed optimal shape and its temperature contours; e) comparison of the initial and optimal shapes; f) decrease in objective function versus the iteration number; g) temperature contours before optimization; h) temperature contours after optimization.

Figure 5-3a depicts the initial meshed physical domain. The temperature contours of the initial domain under the boundary conditions listed in Table 5-1 are shown in Figure 5-3b. It is expected from the temperature contours for the initial shape that the optimal shape should have a boundary surface parallel to the contours to have the required uniform temperature. The optimal meshed shape and its temperature contours shown in Figure 5-3c and d confirm this expectation. The comparison of the initial and optimal shapes is shown in Figure 5-3e. The decrease in objective function with respect to the iteration number is represented in Figure 5-3f. The decrease in objective function is from about 35500 to about 200 in 108 iterations. Since the mesh size is $\left(M,N,L\right)=(15\times15\times15)$, the outer surface has $(M\times L)=15\times15=225$ nodes. Therefore, with regard to Equation (5.5), the square of the difference between 225 nodes and the desired temperature (in this test case, $T_d=50^{\circ}\mathrm{C}$) should be minimized. The final objective function value of 200 confirms the correct implementation of the scheme and its rapid convergence. The plots in Figure 5-3g and h show the temperature contours for outer surface $(AFGD)$

before and after optimization, respectively. It can be seen that the outer surface temperature approaches the desired temperature $T_d = 50^{\circ}\mathrm{C}$ after optimization. By comparing the temperature values before optimization (Figure 5-3g) and the desired temperature $T_d = 50^{\circ}\mathrm{C}$, it can be inferred that the optimal outer surface should be further from the surface subject to the heat flux than the initial outer surface, which is precisely the result seen in Figure 5-3e. The most values of deviation from the desired temperature in the optimal shape of body pertain to the corners of the body. If a fine grid is generated, these values of deviation will be considerably reduced and therefore the objective function value will be decreased further. The results are obtained by a Fortran compiler and computations are run on a PC with Intel Pentium Dual 1.73 and 1G RAM and the computation time is about 28 seconds per iteration or about 50 minutes for the total optimization. The summary of these results are tabulated in Table 5-2.

initial objective function value	~ 35500
final objective function value	~ 200
total iteration number	108
number of nodes on the outer surface	$15 \times 15 = 225$
total optimization time	$\sim 50\,\mathrm{min}$
optimization time per iteration	$\sim 28\,\mathrm{sec}$
variation of temperatures (outer surface) before optimization	$56 - 68$, $T_d = 50^{\circ}\mathrm{C}$
variation of temperatures (outer surface) after optimization	$47.8 - 51$, $T_d = 50^{\circ}\mathrm{C}$

Table 5-2 Summary of results for Test case 1

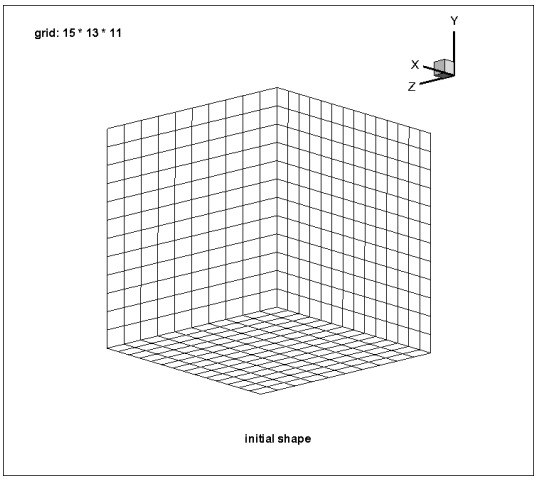
Test case 2

The geometry considered is shown in Figure 5-2 and the numerical values of the coefficients involved in the problem and the coordinates of the physical domain are listed in Table 5-3.

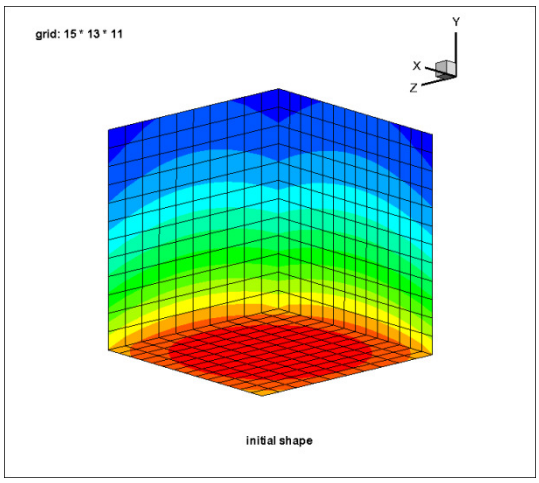
$\dot{q}_{\text{on face BEHC}} = 900\frac{\text{W}}{\text{m}^2}$	$h_{\text{other faces}} = 3\frac{\text{W}}{\text{m}^2.\text{C}}$
$T_{\infty(\text{other faces})} = 30^{\circ}\text{C}$	$k_T = 35\frac{\text{W}}{\text{m.C}}$
$B(1,1,1) = (1,1,1)\text{m}$	$E(M,1,1) = (5,1,1)\text{m}$
$H(M,1,L) = (5,1,5)\text{m}$	$C(1,1,L) = (1,1,5)\text{m}$
$A(1,N,1) = (1,5,1)\text{m}$	$F(M,N,1) = (5,5,1)\text{m}$
$G(M,N,L) = (5,5,5)\text{m}$	$D(1,N,L) = (1,5,5)\text{m}$
$T_d = 65^{\circ}\text{C}$	mesh size=$15\times13\times11$

Table 5-3 Data used for Test case 2

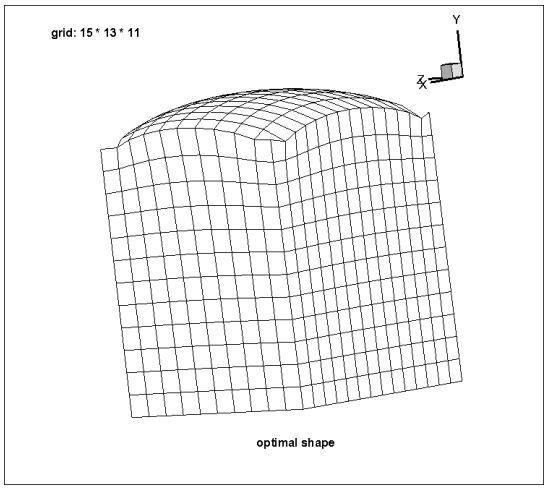
The aim is to find an optimal shape featuring the uniform temperatures on the outer surface $(AFGD)$ for the prescribed boundary conditions given in Table 5-3. The results for Test case 2 are depicted in Figure 5-4.


grid: 15 * 13 * 11
Y
X
Z
initial shape

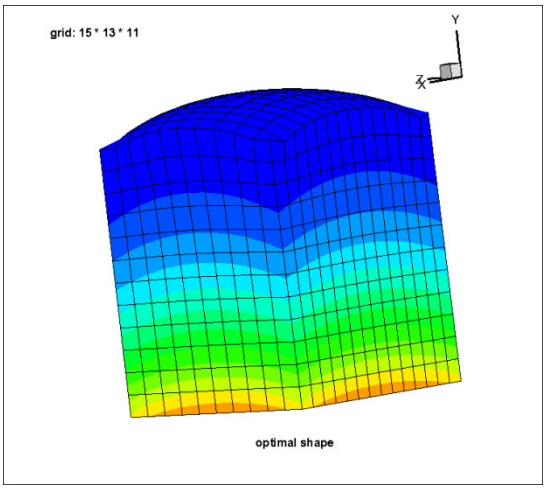
a)


grid: 15 * 13 * 11
Y
X
Z
initial shape

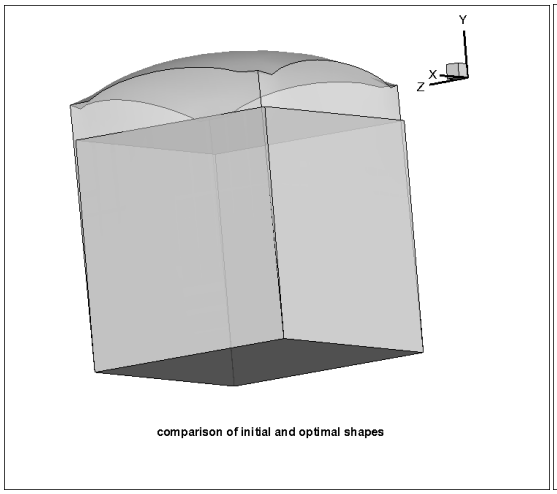
b)


grid: 15 * 13 * 11
Y
Z
X
optimal shape

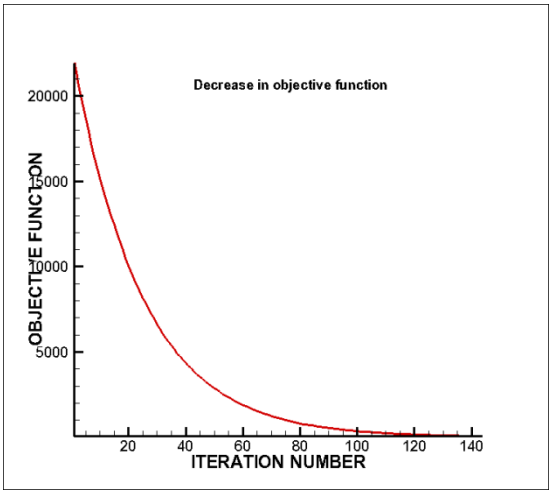
c)


grid: 15 * 13 * 11
Y
Z
X
optimal shape

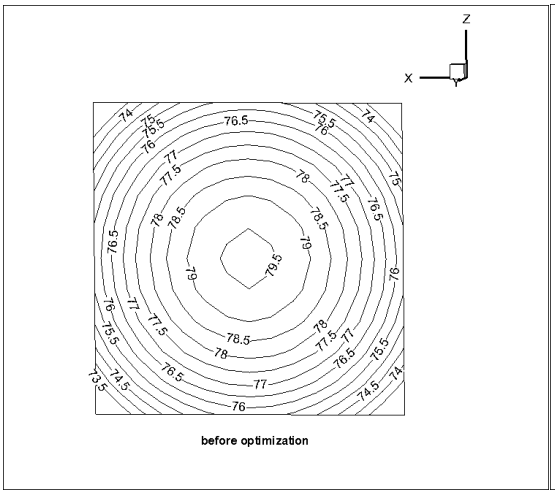
d)



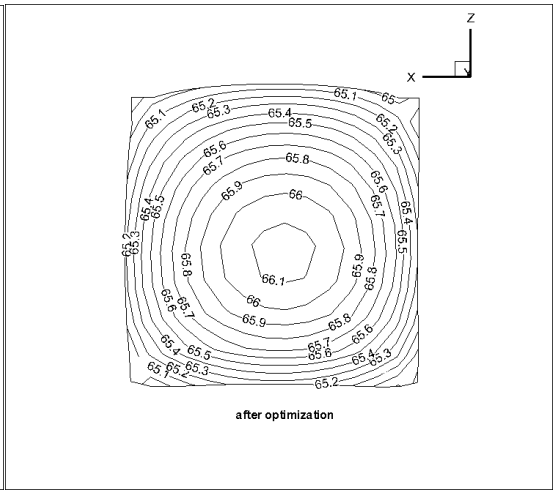
e)



f)



g)



h)

Figure 5-4 a) meshed initial shape; b) meshed initial shape and its temperature contours; c) meshed optimal shape; d) meshed optimal shape and its temperature contours; e) comparison of the initial and optimal shapes; f) decrease in objective function versus the iteration number; g) temperature contours before optimization; h) temperature contours after optimization.

Figure 5-4a depicts the initial meshed physical domain. The temperature contours of the initial domain under the boundary conditions listed in Table 5-3 is shown in Figure 5-4b. It is expected from the temperature contours for the initial shape that the optimal shape should has a boundary surface

parallel to the contours to have the uniform temperature. The optimal meshed shape and its temperature contours shown in Figure 5-4c and d confirm the expectation. The comparison of the initial and optimal shapes is shown in Figure 5-4e. The decrease in objective function with respect to the iteration number is represented in Figure 5-4f. The decrease in objective function is from about 22000 to about 60 in 143 iterations. Since the mesh size is $(M,N,L)=(15,13,11)$, the outer surface has $M\times L=15\times 11=165$ nodes. Therefore, with regard to Equation (5.5), the square of the difference between 165 nodes and the desired temperature (in this test case, $T_d=65^{\circ}\mathrm{C}$) should be minimized. The final objective function value of 60 shows the accuracy of the implemented scheme. The plots in Figure 5-4g and h show the temperature contours for outer surface $(AFGD)$ before and after optimization, respectively. It can be seen that the outer surface temperature approaches the desired temperature $T_d=65^{\circ}\mathrm{C}$ after optimization. By comparing the temperature values before optimization (Figure 5-4g) and the desired temperature $T_d=65^{\circ}\mathrm{C}$ it can be inferred that the optimal outer surface should be further from the surface involving the heat flux than the initial outer surface, which is depicted in Figure 5-4e. The computation time is about 14 seconds per iteration or about 33 minutes for the total optimization. The summary of these results are tabulated in Table 5-4.

initial objective function value	~ 22000
final objective function value	~ 60
total iteration number	143
number of nodes on the outer surface	$15\times 11=165$
total optimization time	$\sim 33\,\mathrm{min}$
optimization time per iteration	$\sim 14\,\mathrm{sec}$
variation of temperatures (outer surface) before optimization	$73.5-79.5$, $T_d=65^{\circ}\mathrm{C}$
variation of temperatures (outer surface) after optimization	$65-66.1$, $T_d=65^{\circ}\mathrm{C}$

Table 5-4 Summary of results for Test case 2

Test case 3

For the third test case, consider the physical domain shown in Figure 5-5. The semicircular surface $BEHC$ is subject to a prescribed heat flux $\dot{q}$ and the other faces are exposed to the convective heat transfer. The numerical values for the coefficients involved in the problem and the coordinates of the physical domain are listed in Table 5-5.

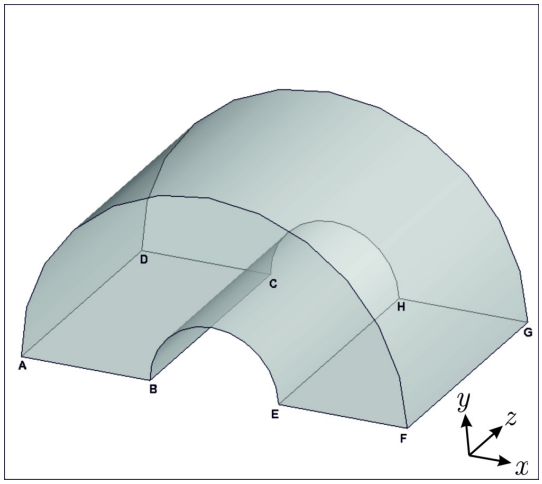
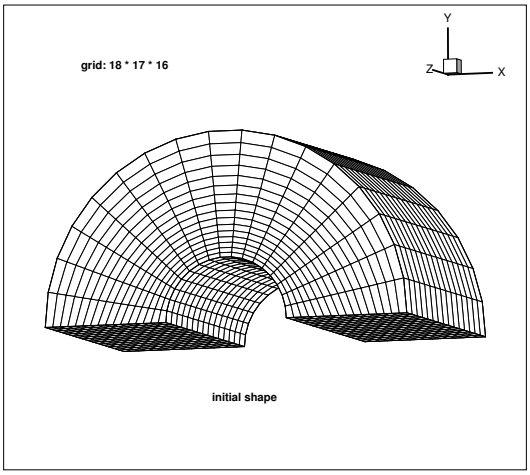


Figure 5-5 The heat conducting body subject to the prescribed boundary conditions.

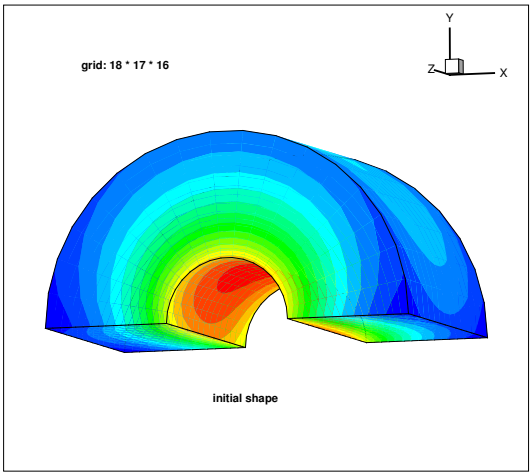
$\dot{q}_{\text{on face BEHC}} = 700\dfrac{\text{W}}{\text{m}^2}$	$h_{\text{other faces}} = 5\dfrac{\text{W}}{\text{m}^2.\text{C}}$
$T_{\infty(\text{other faces})} = 40^{\circ}\text{C}$	$k_T = 125\dfrac{\text{W}}{\text{m.C}}$
$B(1,1,1) = (5,1,1)\text{m}$	$E(M,1,1) = (15,1,1)\text{m}$
$H(M,1,L) = (15,1,20)\text{m}$	$C(1,1,L) = (5,1,20)\text{m}$
$A(1,N,1) = (-5,1,1)\text{m}$	$F(M,N,1) = (25,1,1)\text{m}$
$G(M,N,L) = (25,1,20)\text{m}$	$D(1,N,L) = (-5,1,20)\text{m}$
$T_d = 55^{\circ}\text{C}$	mesh size=$18\times17\times16$

Table 5-5 Data used for Test case 3

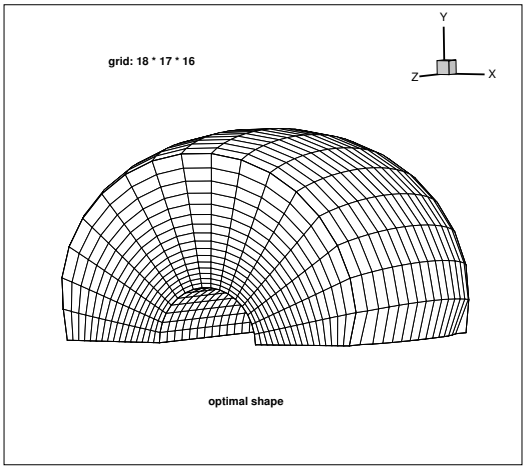
The aim is to find an optimal shape featuring the uniform temperatures on the semicircular outer surface $(AFGD)$ for the prescribed boundary conditions given in Table 5-5.



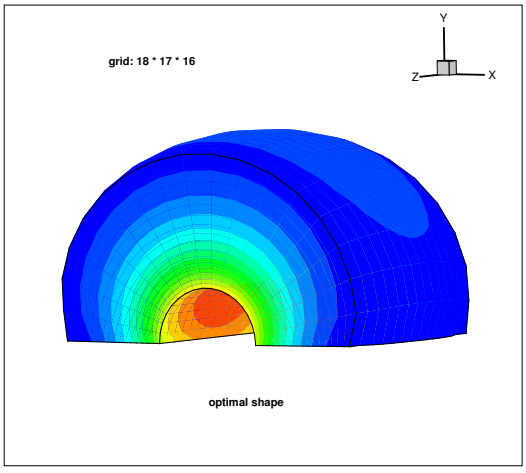
a)



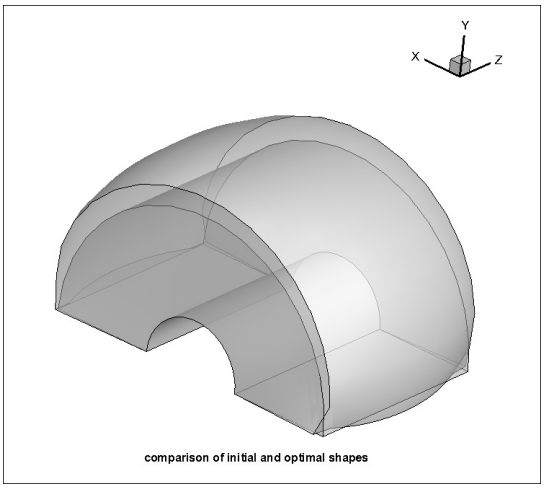
b)



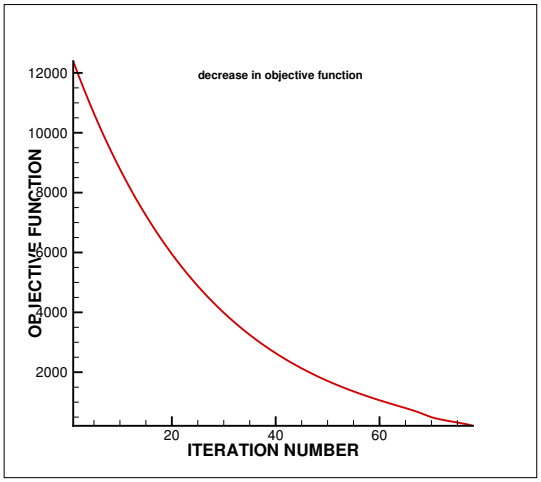
c)



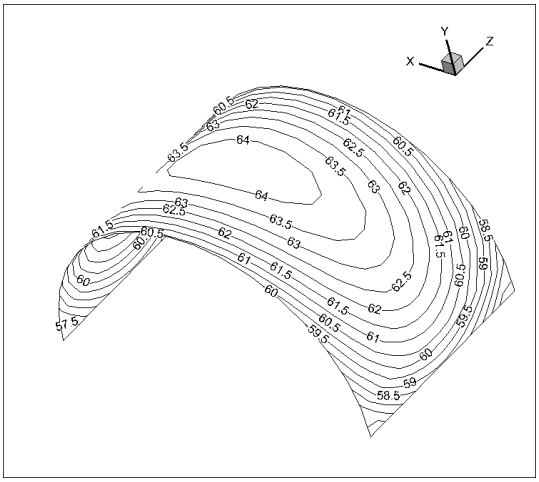
d)



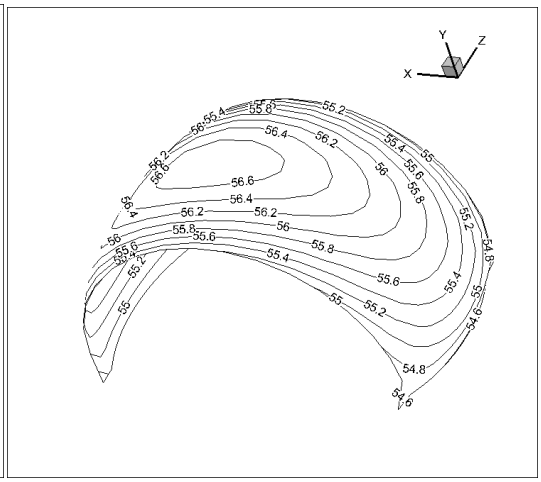
e)



f)



g)



h)

Figure 5-6 a) meshed initial shape; b) meshed initial shape and its temperature contours; c) meshed optimal shape; d) meshed optimal shape and its temperature contours; e) comparison of the initial and optimal shapes; f) decrease in objective function versus the iteration number; g) temperature contours before optimization; h) temperature contours after optimization.

The summary of the results are tabulated in Table 5-6.

initial objective function value	~ 12390
final objective function value	~ 210
total iteration number	78
number of nodes on the outer surface	$18 \times 16 = 288$
total optimization time	$\sim 50\,\text{min}$
optimization time per iteration	$\sim 38\,\text{sec}$
variation of temperatures (outer surface) before optimization	$57.5 - 64$, $T_d = 55^{\circ}\text{C}$
variation of temperatures (outer surface) after optimization	$54.6 - 56.6$, $T_d = 55^{\circ}\text{C}$

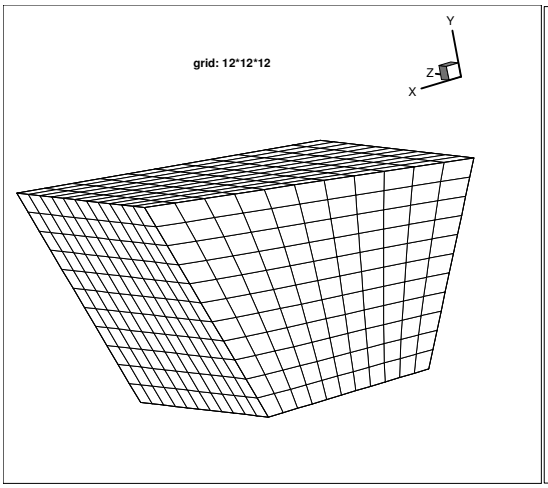
Table 5-6 Summary of results for Test case 3

Test case 4

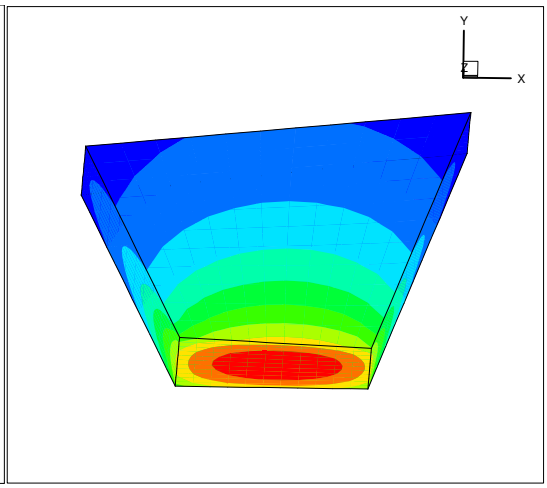
In this test case, an irregular shape is considered. The data for the test case is given in Table 5-7.

$\dot{q}_{\text{on face BEHC}} = 400 \dfrac{\text{W}}{\text{m}^2}$	$h_{\text{other faces}} = 7 \dfrac{\text{W}}{\text{m}^2.\text{C}}$
$T_{\infty(\text{other faces})} = 40^{\circ}\text{C}$	$k_T = 20 \dfrac{\text{W}}{\text{m.C}}$
$B(1,1,1) = (1,1,1)\text{m}$	$E(M,1,1) = (5,1,1)\text{m}$
$H(M,1,L) = (5,1,6)\text{m}$	$C(1,1,L) = (1,1,7)\text{m}$
$A(1,N,1) = (-1,5,1)\text{m}$	$F(M,N,1) = (7,6,1)\text{m}$
$G(M,N,L) = (7,6,6)\text{m}$	$D(1,N,L) = (-1,5,7)\text{m}$
$T_d = 42^{\circ}\text{C}$	mesh size=$12 \times 12 \times 12$

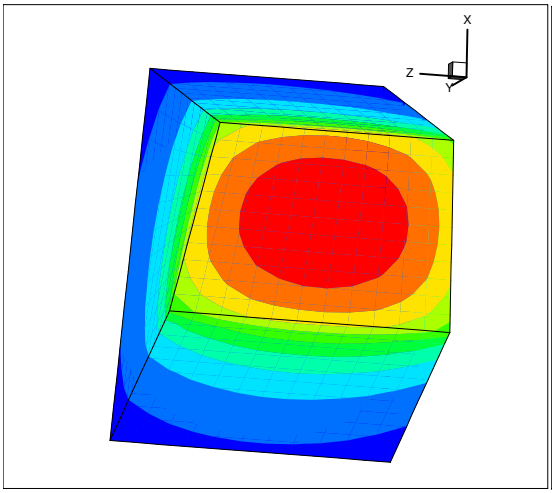
Table 5-7 Data used for Test case 4



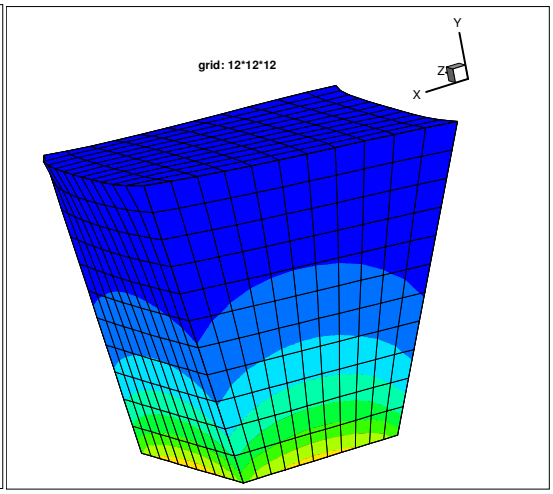
a) Initial shape mesh



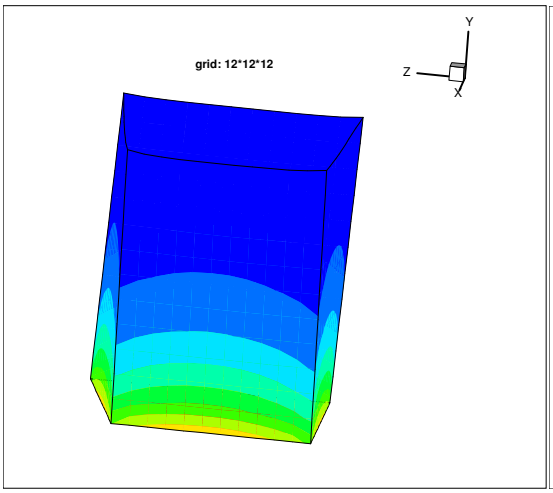
b) Initial shape temperature



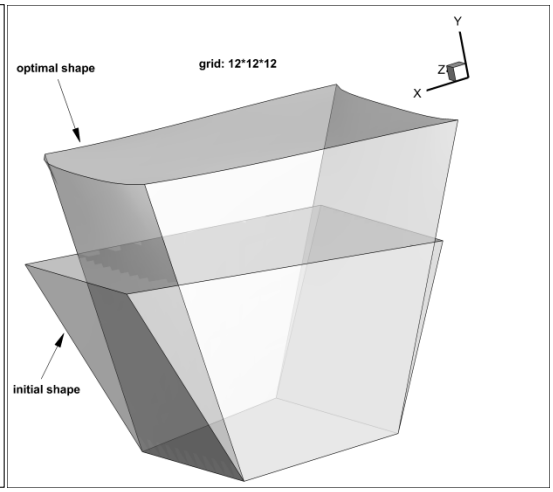
c) Initial shape temperature



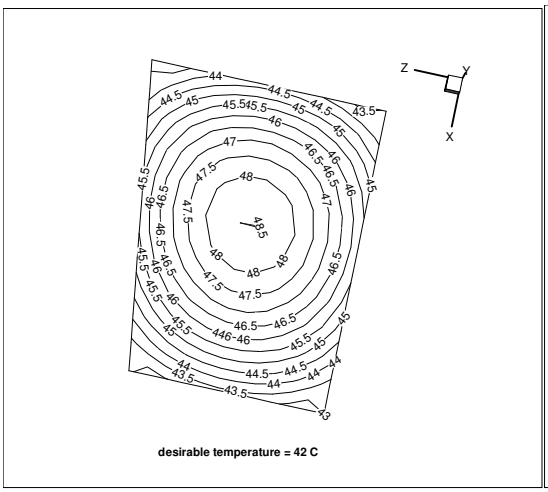
d) Optimal shape temperature



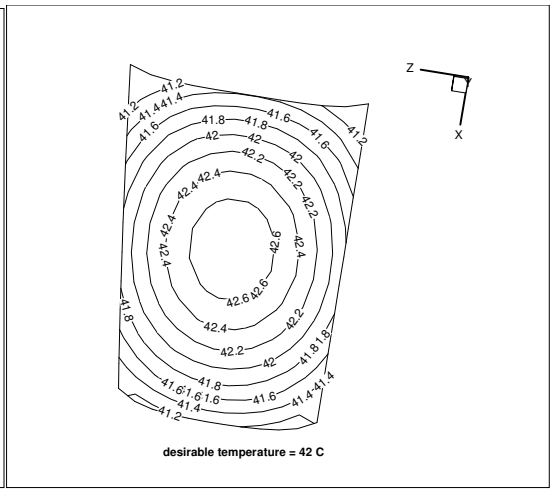
e) Optimal shape temperature



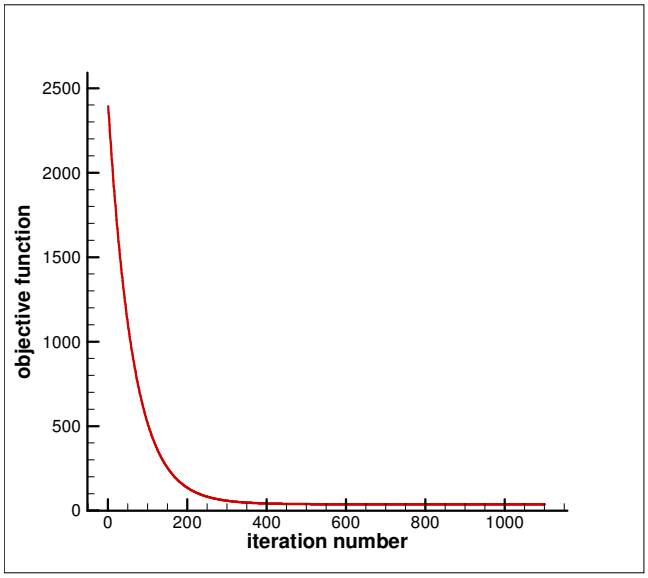
f) Comparison of the initial and optimal shapes



g) Initial shape outer surface



h) Optimal shape outer surface



i) Objective function versus iteration number

Figure 5-7 Results for Test case 4

Decrease in objective function	From ~2393 to ~36
Percentage of decrease in objective function	%98.5
Computation time	10 min : 50 seconds
Number of iterations	1100

Table 5-8 Summary of results for Test case 4

Moreover, three different grid sizes are considered for this test case to study the effect of the grid size on the optimization process and the optimal shape. The result is shown in Figure 5-8. The plot shows that the optimal shape is approximately independent of the grid size.

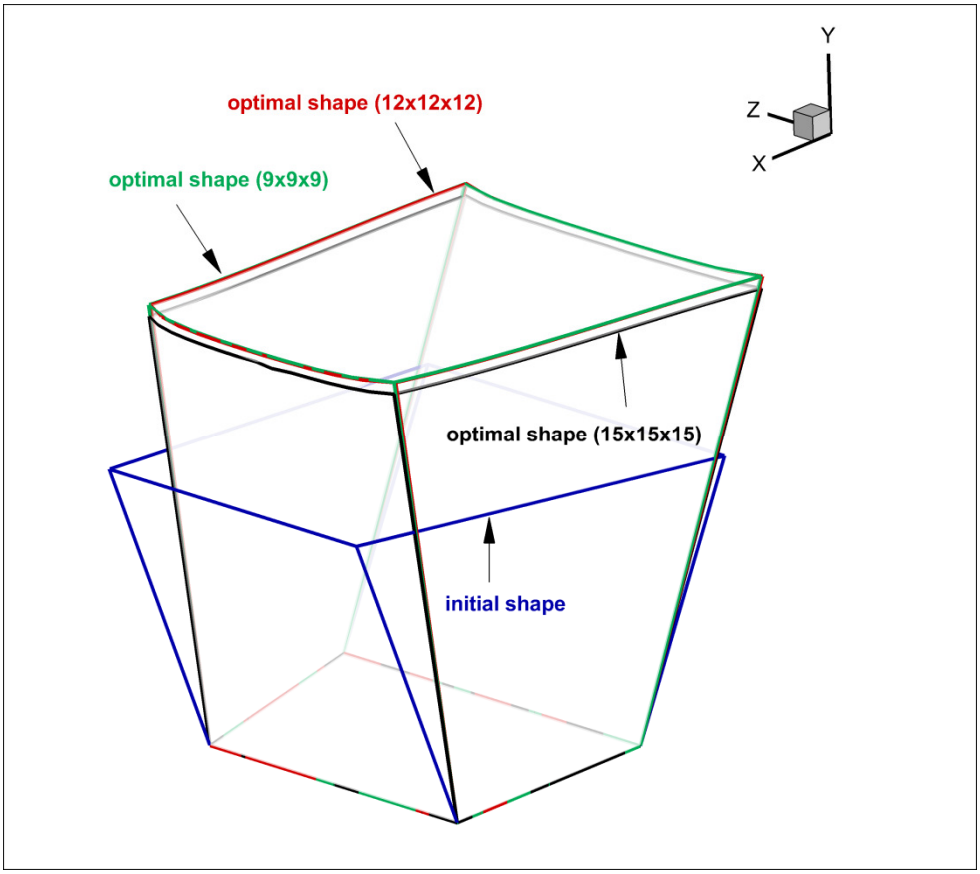


Figure 5-8 Comparison of the initial and optimal shapes for three different grid sizes. The plot shows that the optimal shape is approximately independent of the grid size.

Test case 5 (different initial shapes)

In this test case, two regular shapes are considered as initial shapes. The data for the test case is given in Table 5-9.

$\dot{q}_{\text{on face BEHC}} = 1300 \frac{\text{W}}{\text{m}^2}$	$h_{\text{other faces}} = 8 \frac{\text{W}}{\text{m}^2.\text{C}}$
$T_{\infty(\text{other faces})} = 30^{\circ}\text{C}$	$k_T = 125 \frac{\text{W}}{\text{m.C}}$
$B(1,1,1) = (1,1,1)\text{m\&}(1,1,1)\text{m}$	$E(M,1,1) = (5,1,1)\text{m\&}(5,1,1)\text{m}$
$H(M,1,L) = (5,1,5)\text{m\&}(5,1,5)\text{m}$	$C(1,1,L) = (1,1,5)\text{m\&}(1,1,5)\text{m}$
$A(1,N,1) = (1,5,1)\text{m\&}(1,7,1)\text{m}$	$F(M,N,1) = (5,5,1)\text{m\&}(5,7,1)\text{m}$
$G(M,N,L) = (5,5,5)\text{m\&}(5,7,5)\text{m}$	$D(1,N,L) = (1,5,5)\text{m\&}(1,7,5)\text{m}$
$T_d = 50^{\circ}\text{C}$	mesh size=$10 \times 10 \times 10$

Table 5-9 Data used for Test case 5

Initial shape #1: As given in Table 5-9, $y-$component of the initial shape outer surface is 5m. By comparing the outer surface temperatures shown in Figure 5-10 and the desired temperature $T_d = 50^{\circ}\text{C}$ we can predict that the development of the shape in $y-$direction should be away from the surface $BEHC$ (the surface with a heat flux). This is shown in Figure 5-9 and the outer surface temperatures for the optimal shape are shown in Figure 5-11.

Initial shape #2: For this initial shape, the $y-$component of the initial shape outer surface is 7m. By comparing the outer surface temperatures shown in Figure 5-12 and the desired temperature $T_d = 50^{\circ}\text{C}$ we can predict that the development of the shape in $y-$direction should be toward the surface $BEHC$. The outer surface temperatures for the optimal shape are shown in Figure 5-13. A summary of the results is given in Table 5-10.

	Initial shape #1	Initial shape #2
Decrease in objective function	From 4626.9 to 35.3	From ~2025.0 to ~34.6
Percentage of decrease in objective function	%99.2	%98.3
Number of iterations	90	148

Table 5-10 Summary of results for Test case 5

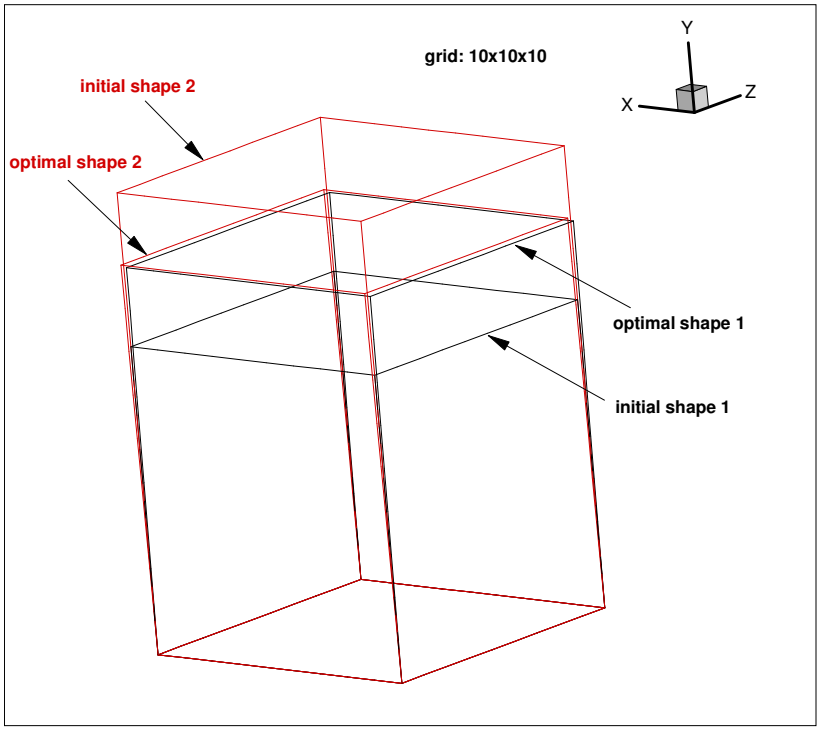


Figure 5-9 Comparison of the initial and optimal shapes for two different initial shapes. The plot shows that the optimal shape is approximately independent of the initial shape.

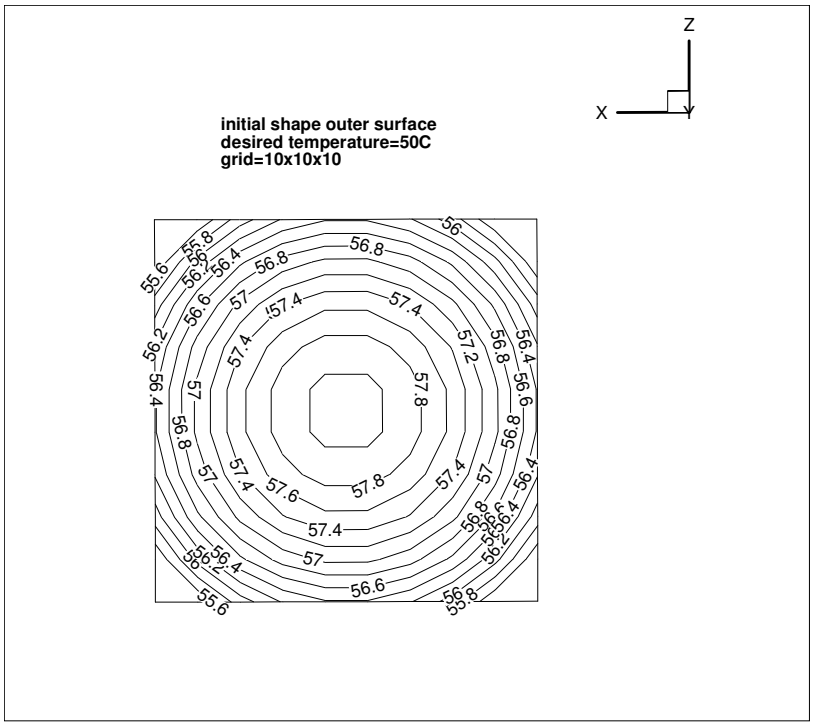


Figure 5-10 Outer surface temperatures for the initial shape #1.

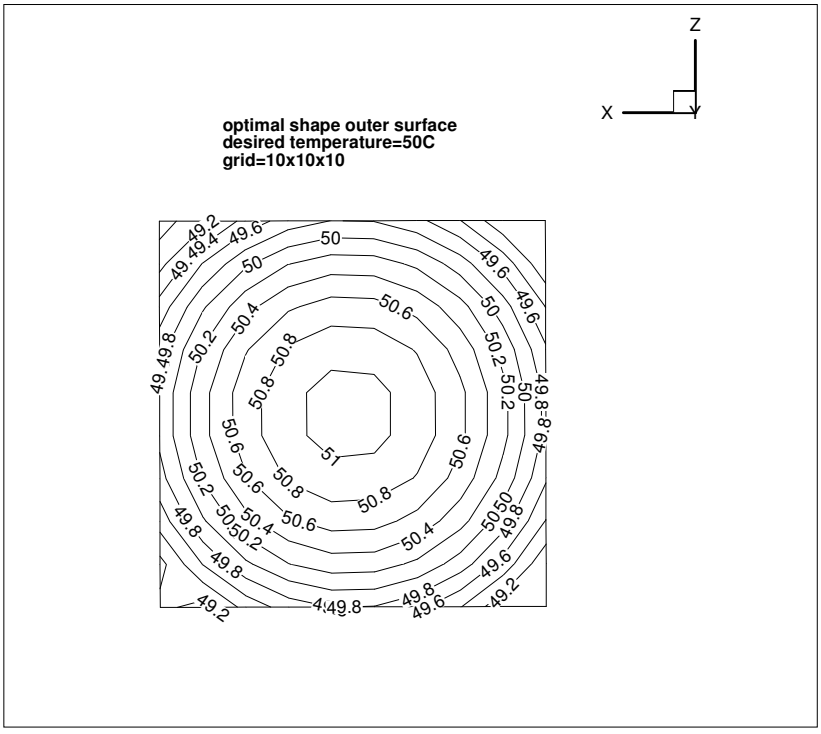


Figure 5-11 Outer surface temperatures for the optimal shape (for the initial shape #1).

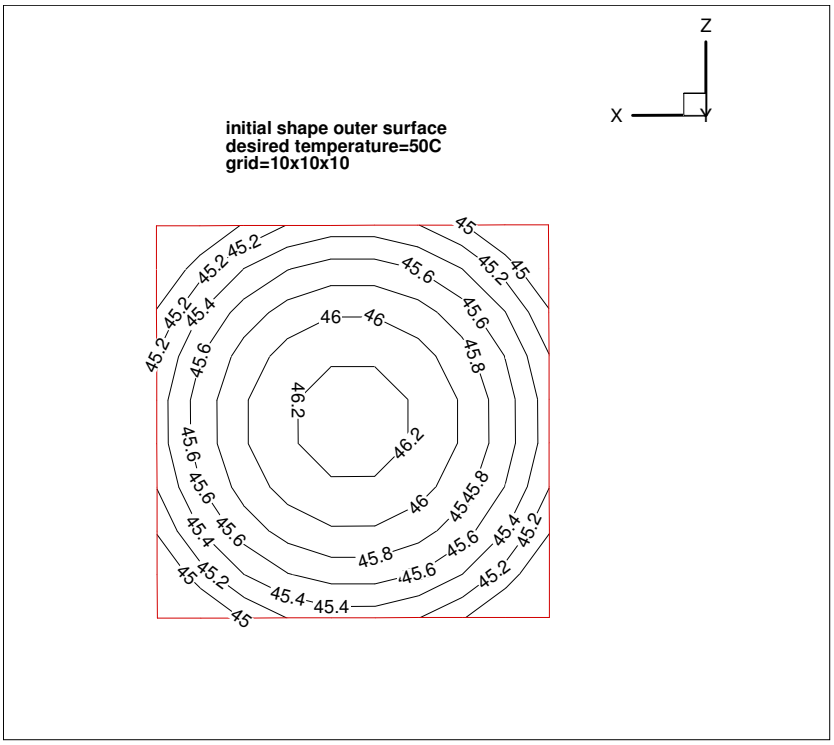


Figure 5-12 Outer surface temperatures for the initial shape #2.

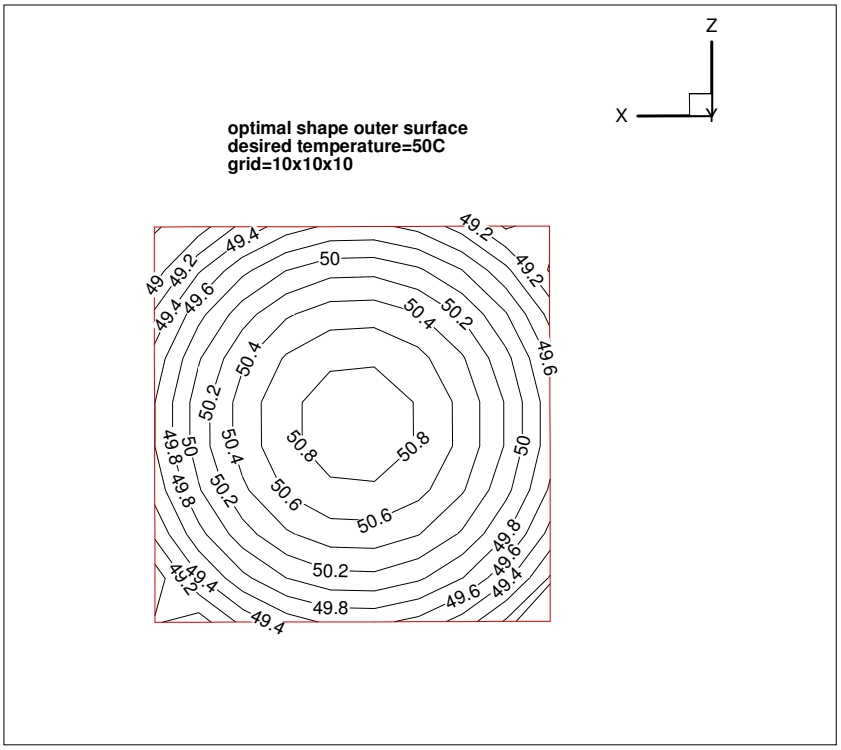


Figure 5-13 Outer surface temperatures for the optimal shape (for the initial shape #2).

5.6 Conclusion

This chapter addressed the optimal shape design for some 3-D heat transfer problems with combinations of the Neumann and Robin boundary conditions. The scheme is based on mapping a 3-D irregular physical domain into a 3-D regular (cuboid) computational domain and takes advantage of the discretization of the governing equation (Laplace Equation) using the finite difference method. One of the main contributions of this study is the derivation of an explicit expression for the sensitivity allowed by the mapping between the physical and computational domains. Unlike the perturbation method which needs the high number of additional solutions of the direct heat transfer problem per iteration, the proposed method takes advantage of only one solution of the direct problem per iteration. This results in accurate results and considerably decreases the computation time. The decrease in objective function and matching the temperature of the nodes on the boundary of the physical domain and desired one showed in test cases revealed the accuracy of the scheme.

Part two: Shape Optimization in Aerodynamics

6. Airfoil shape optimization

6.1 Introduction

So far, we have used the optimal shape design scheme for the heat conduction problems in which the Laplace equation was satisfied. In this chapter, we focus on aerodynamic shape optimization which is also governed by the Laplace equation.

By the advent of the modern high speed computers over the last few decades, Computational Fluid Dynamics (CFD) has been extensively employed as an analysis tool and as a design optimization tool. As was shown in preceding chapters, among the methodologies often employed in shape optimization are gradient-based techniques, in which a specified objective function is minimized. In airfoil shape optimization, the objective function can be, say, a measure of the drag on the airfoil surface and it would be desirable to reduce the drag on the airfoil surface as much as possible. Aerodynamic shape optimization can be studied based on many factors including the type of flow regime (subsonic, transonic etc), the value of the Angle of Attack (AOA), the type of the governing equations (Laplace, Euler, Navier-Stokes), the type of grid generation method (structured or unstructured, if structured: O-type, C-type, H-type), the method of discretization of the governing equations (FDM, FEM, FVM), the type of optimization method (Newton, conjugate gradient, quasi-Newton algorithms, genetic algorithm etc), the type of shape sensitivity analysis (finite difference, adjoint method, automatic differentiation etc), type of the body being optimized (wing, fuselage, blade etc), and type of objective function. The detailed study of an aerodynamic shape optimization problem considering all of these factors is beyond the scope of this thesis. In this chapter, we restrict our study only to the 2-D shape optimization of an airfoil in an irrotational and incompressible flow governed by the Laplace equation. The procedure employed in previous chapters, which was based on elliptic grid generation, a novel sensitivity

analysis (based on finite difference method), and an optimization method, will also be used in the airfoil shape optimization problem of interest in this chapter. In addition to the conjugate gradient method, an efficient version of quasi-Newton method (BFGS) will also be used as an optimization algorithm.

Some of the earliest studies using a combination of CFD with numerical optimization in aerodynamic were made by Hicks and Henne [103, 104]. In [103], a procedure for optimal design of symmetric low-drag, nonlifting transonic airfoils in inviscid flow is proposed. The proposed procedure uses an optimization program based on the method of the feasible directions coupled with an analysis program that utilizes a relaxation method to solve the partial differential equation that governs the inviscid, transonic, small disturbance fluid flow. The drag minimization with geometric constraints is considered in this reference. In fluid dynamics, Pironneau was the first one to use the adjoint equations for design [105]. This is the first application of control theory to design optimization. However, within the field of aeronautical computational fluid dynamics, Jameson was the first researcher who used the continuous adjoint formulation for aerodynamic shape optimization in transonic potential flows and flows governed by Euler equations [106-109]. Giles made considerable contributions to the development of the discrete adjoint approach [110-113]. In [112], the adjoint equations are formulated for the transonic design applications for which there are shocks. The adjoint equations were already formulated for the incompressible or subsonic flows in which the assumption that the original nonlinear flow solution is smooth is valid. In [113], a number of algorithm developments are presented for adjoint methods using the "discrete" approach. In continuous adjoint method, the original partial differential equations are linearized, the adjoint partial differential equation and appropriate boundary conditions are formulated and finally the equations are discretized. Unlike the continuous adjoint approach, in discrete adjoint approach the partial differential equations are discretized, the discrete equations are linearized, and then the transpose of the linear operator is used to form the adjoint problem. The adjoint equations have also been used by Baysal and Eleshaky to infer the optimal design for a scramjet - afterbody configuration which yields the maximum axial thrust [114] and by Ta'asan, Kuruvila and Salas to obtain an optimal airfoil shape [115]. Baysal and Eleshaky's work was based on a computational fluid dynamics - sensitivity analysis algorithm (two different quasi-analytical approaches: the direct method and the adjoint variable

method) to solve Euler equations for the inviscid analysis of the flow. Adjoint methods have been applied to incompressible viscous flow problems by Cabuk, Shung and Modi [116] and Desai and Ito [117]. Cabuk et al. worked on the problem of determining the profile of a channel or duct that provides the maximum static pressure rise by solving the incompressible, laminar flow governed by the steady state Navier-Stokes equations. Early applications of discrete adjoint methods on unstructured meshes can be found in works by Elliot and Peraire in inviscid [118] and viscous flows [119] for 2D and 3D [120] configurations. In [118], an inverse design procedure for single- and multi-element airfoils using unstructured grids and based on the Euler equations are presented. The discrete adjoint method is used to compute the sensitivities and the results are compared with corresponding finite difference values. It is shown that the use of the adjoint method practically eliminates the dependence of the objective function gradient computation on the number of design variables. The continuous adjoint approach for unstructured grids has been developed by Anderson [121, 122]. In [121], aerodynamic shape optimization on unstructured grids using a continuous adjoint approach is developed and analyzed for inviscid and viscous flows. B spline and Bezier curves are employed to parameterize the airfoil surface. The objective functions considered include drag minimization, lift maximization, and matching a specified pressure distribution. The quasi-Newton optimization method is used to obtain the optimal design. Evolutionary algorithms, as methods that do not need the computation of the gradient, have recently gained much attention in the context of aerodynamic shape optimization [123-127]. Although they are of extremely high computational cost, they have the advantage that they can escape from a "local minimum" (a major issue in using gradient based methods) and have the ability of finding globally optimum solutions amongst many local optima [128, 129]. A detailed study of many methods in shape optimization in fluid mechanics is given by Mohammadi and Pironneau [130].

The adjoint approach, as an alternative to the finite difference method to compute the gradient of functional with respect to the design variables, is computationally very efficient. Therefore, as far as the computational cost is concerned, it is the appropriate choice. This is the case when there is a large number of design variables which makes use of the finite difference method impractical. The differences between the adjoint method and the finite difference one (to compute the gradient of functional with respect to the design variables) can be summarized below

Adjoint method: N Design Variables, 1 Flow Solution, 1 Adjoint Calculation

Finite difference method: N Design Variables, N Flow Solutions. Because

$$\frac{\partial \mathcal{J}}{\partial \alpha_j} = \frac{\left[\mathcal{J}(\alpha_j + \delta\alpha_j) - \mathcal{J}(\alpha_j)\right]}{\delta\alpha_j}$$

where $\mathcal{J}$ is the objective function and α_j are the design variables [131]. As can be seen, the aerodynamic shape optimization with a large number of design variables seems to be computationally practical only when the adjoint method is used. However, as will be shown in this chapter, a novel sensitivity analysis will be presented (the same methodology as heat transfer case in previous chapters) which makes use of the finite difference method comparable (from the computational cost viewpoint) to the use of adjoint method.

Like shape optimization in heat transfer, aerodynamic shape optimization being considered in this chapter consists of three steps, namely, grid generation and flow equation solver to find the pressure on the airfoil surface, sensitivity analysis to compute the gradient of the objective function with respect to the design variables, and an optimization method to minimize the functional and reach optimum solution. Since the methodology used here is similar to one used in Chapters 2 and 3, less description will be given in grid generation and optimization steps unless new concepts (airfoil parameterization using Bezier curves and optimization using BFGS method) are introduced.

6.2 Governing equation for irrotational, incompressible flow: Laplace equation

Principle of mass conservation for an incompressible flow gives

$$\nabla \cdot \mathbf{V} = 0 \qquad \text{(continuity equation)} \tag{6.1}$$

For an irrotational flow, we have

$$\boldsymbol{\xi} = \nabla \times \mathbf{V} = 0 \tag{6.2}$$

where $\boldsymbol{\xi}$ is the vorticity vector. Considering the following vector identity

$$\nabla \times (\nabla \phi) = 0 \tag{6.3}$$

where ϕ is a scalar function (named velocity potential), then we get

$$\mathbf{V} = \nabla \phi \tag{6.4}$$

Comparing Equations (6.1) and (6.4), we obtain

$$\nabla \cdot (\nabla \phi) = 0 \tag{6.5}$$

or

$$\nabla^2 \phi = 0 \qquad \text{(Laplace Equation)} \tag{6.6}$$

For a two dimensional incompressible flow, a stream function ψ can be defined such that

$$u = \frac{\partial \psi}{\partial y} \tag{6.7}$$

$$v = -\frac{\partial \psi}{\partial x} \tag{6.8}$$

where u and v are the components of the velocity vector $\mathbf{V}$, i.e, $\mathbf{V} = u\mathbf{i} + v\mathbf{j}$ ($\mathbf{i}$ and $\mathbf{j}$ are the unit vectors in x and y directions, respectively). If incompressible flow is irrotational, from the irrotationality condition we have

$$\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = 0 \tag{6.9}$$

Substituting Equations (6.7) and (6.8) into Equation (6.9), we have

$$\frac{\partial}{\partial x}(-\frac{\partial \psi}{\partial x}) - \frac{\partial}{\partial y}(\frac{\partial \psi}{\partial y}) = 0 \tag{6.10}$$

or

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0 \tag{6.11}$$

which is the Laplace equation. We can conclude that any irrotational, incompressible flow has a velocity potential and stream function (for two-dimensional flow) that both satisfy Laplace equation [132].

Before proceeding further, consider irrotational incompressible flow over an airfoil (Figure 6-1). The boundary conditions are as below

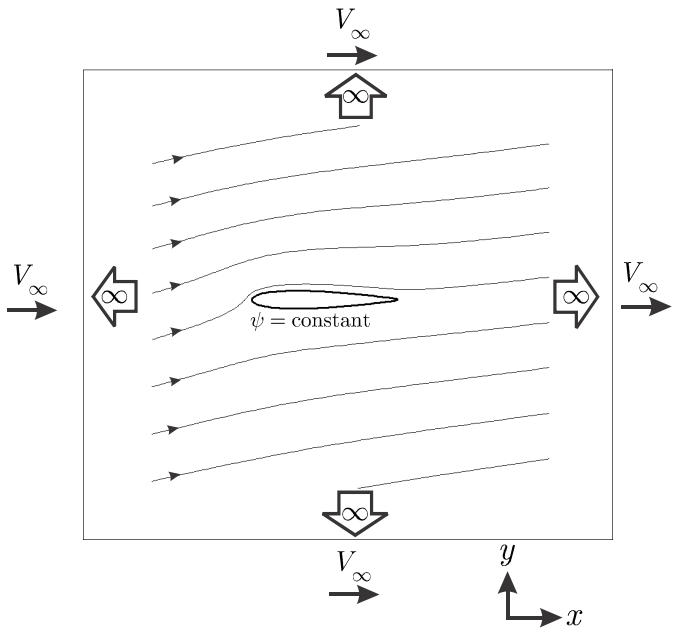


Figure 6-1 Boundary conditions at infinity and on the airfoil surface (no-penetration). The flow is inviscid.

Conditions at infinity: Far away from the airfoil surface (toward infinity), in all directions, the flow approaches the uniform free stream conditions. If the angle of attack (AOA) is α, the free stream velocity V_∞, the components of the flow velocity can be written as

$$u = \frac{\partial \psi}{\partial y} = V_\infty \cos\alpha \tag{6.12}$$

$$v = -\frac{\partial \psi}{\partial x} = V_\infty \sin\alpha \tag{6.13}$$

Condition on the airfoil surface: The relevant boundary condition at the airfoil surface for this inviscid flow is the no-penetration boundary condition. Thus the velocity vector must be *tangent* to the surface. This wall boundary condition can be expressed by

$$\frac{\partial \psi}{\partial s} = 0 \qquad \text{or} \qquad \psi = \text{constant} \tag{6.14}$$

where s is tangent to the surface. If the flow is tangent to the airfoil surface, the component of the velocity normal to the airfoil surface must be zero. Let $\mathbf{n}$ be a unit vector normal to the surface, the wall boundary condition can also be written as

$$\mathbf{V} \cdot \mathbf{n} = (\nabla \phi) \cdot \mathbf{n} = 0 \tag{6.15}$$

or

$$\frac{\partial \phi}{\partial n} = 0 \tag{6.16}$$

In the problem of the flow over an airfoil, if the free stream velocity and the angle of attack are known, from the boundary conditions at infinity (Equations (6.12) and (6.13)) and the wall boundary condition (Equation (6.14)) one can compute the stream function ψ at any point of the physical domain (flow region). Then, by knowing ψ one can compute the velocity of all points. Since for an incompressible flow, the pressure coefficient is a function of the velocity only, one can obtain the pressure of any point in the flow region, as will be shown.

Pressure coefficient: The pressure coefficient C_p is defined as

$$C_p \equiv \frac{p - p_\infty}{\frac{1}{2}\rho_\infty V_\infty^2} \tag{6.17}$$

At standard sea level conditions,

$$\rho_\infty = 1.23 \frac{\text{kg}}{\text{m}^3}, p_\infty = 1.01 \times 10^5 \frac{\text{N}}{\text{m}^2} \tag{6.18}$$

As stated before, for an incompressible flow, the pressure coefficient can be expressed in terms of the velocity only. Consider an airfoil which is immersed in an incompressible flow. The free stream pressure and velocity are p_∞ and V_∞, respectively. By applying Bernoulli equation for an arbitrary point where pressure and velocity are p and V, respectively, we have

$$p_\infty + \frac{1}{2}\rho V_\infty^2 = p + \frac{1}{2}\rho V^2 \tag{6.19}$$

or

$$p - p_\infty = \frac{1}{2}\rho(V_\infty^2 - V^2) \tag{6.20}$$

Substituting Equation(6.20) into Equation (6.17), we have

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho V_\infty^2} = \frac{\frac{1}{2}\rho(V_\infty^2 - V^2)}{\frac{1}{2}\rho V_\infty^2} = 1 - (\frac{V}{V_\infty})^2 \tag{6.21}$$

6.3 Grid generation

We have presented all relations needed to obtain the pressure distribution in incompressible, irrotational, inviscid flow over an airfoil. To calculate the pressure of any point in the flow region, a grid should be generated over the region. The elliptic grid generation used in two dimensional heat transfer problem will be employed. The type of the elliptic grid generation used here is O-type which results in a smooth and orthogonal grid over the airfoil surface. The O-type grid generation technique has the advantage that the grids around the airfoil are orthogonal. The discretization of the physical domain and the corresponding computational domain are shown in Figure 6-2 and Figure 6-3, respectively. In the computational domain, M and $N = 2N_1 + 2N_2 + N_3 + 6$ are the number of nodes in the ξ and η directions, respectively.

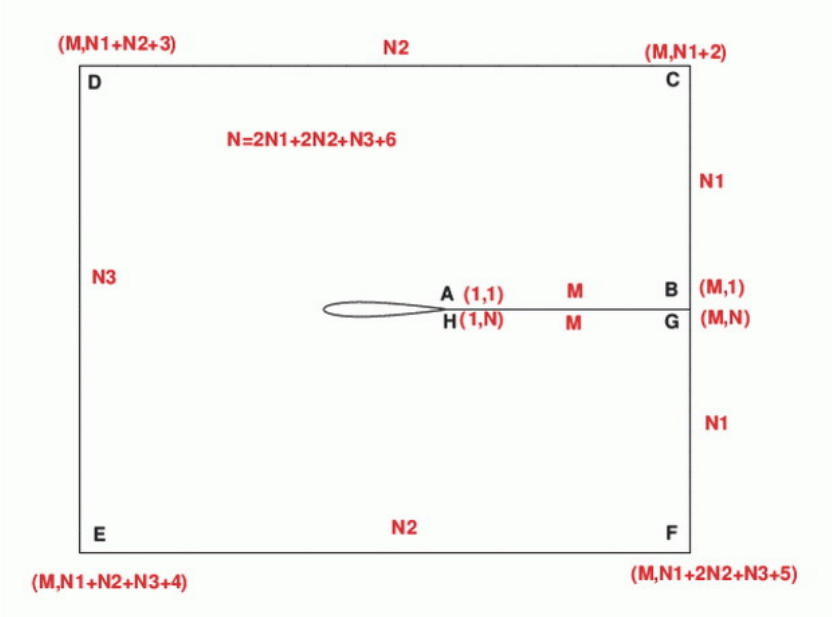


Figure 6-2 Physical domain showing the discretization of the boundaries used for O-type elliptic grid generation technique.

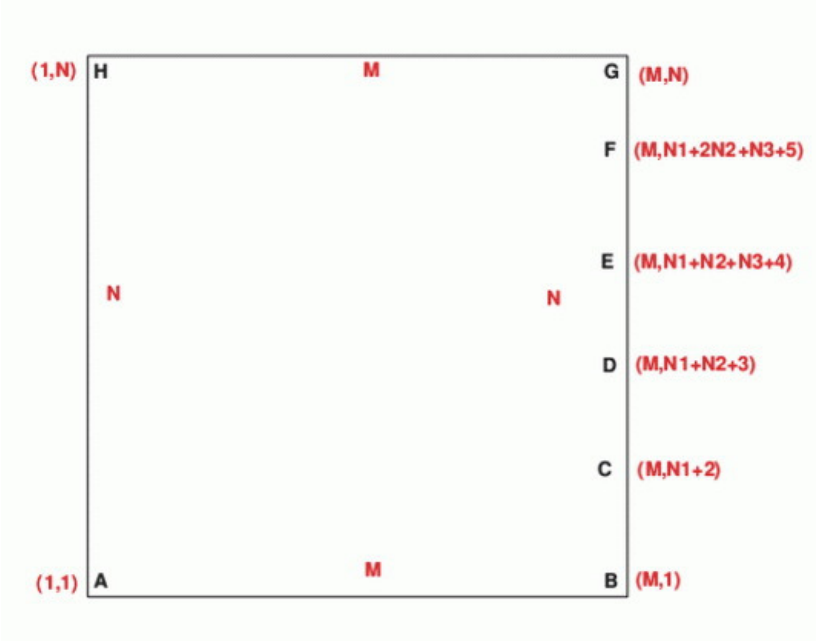
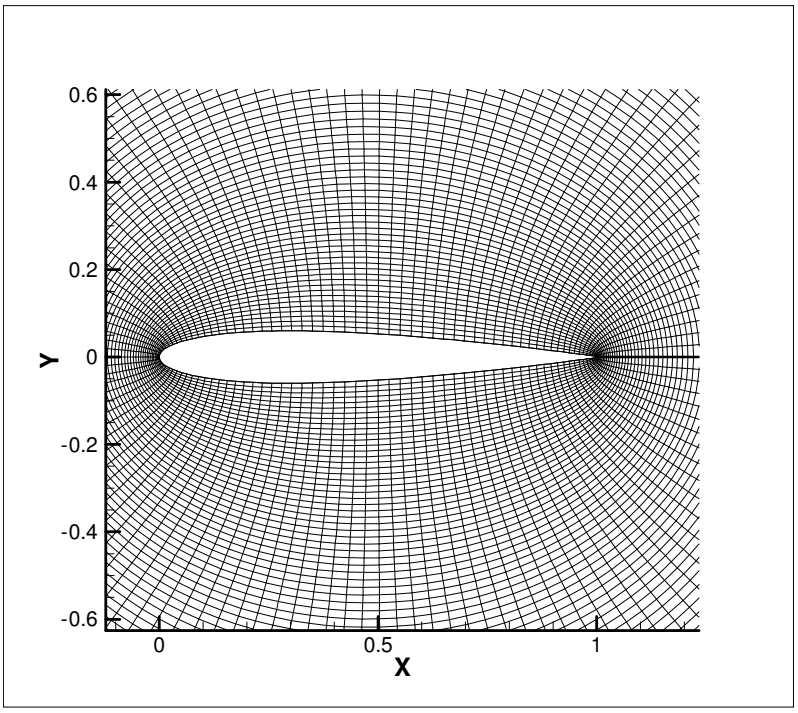
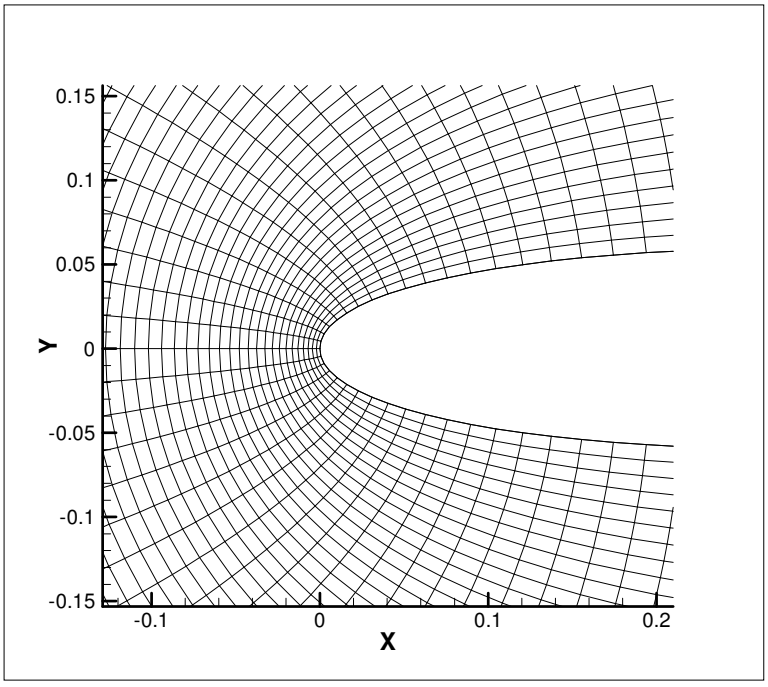


Figure 6-3 Computational domain showing the discretization of the physical domain boundaries

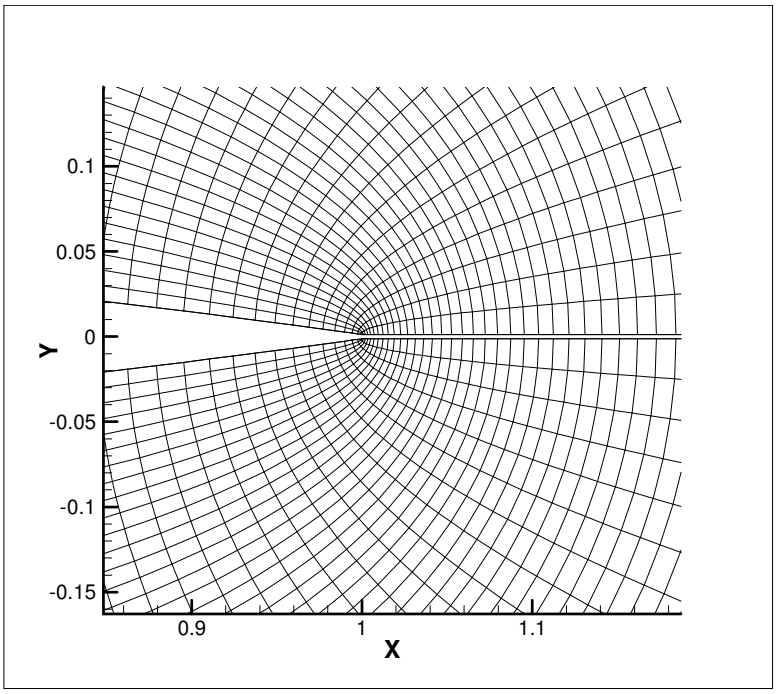
The resulting O-grid scheme over an airfoil is shown in Figure 6-4.



Close-up view of O-type grid around the airfoil.



Magnified view of grid around the leading edge.



Magnified view of grid around the trailing edge.

Figure 6-4 O-type grid (elliptic) around an airfoil. This close-up view of the grid shows orthogonality and smoothness of the gridlines especially near airfoil surface.

The initial guess for the elliptic grid generation is performed using the TFI method. Since the TFI method is an algebraic technique and does not require much computational time, it will be an appropriate initial guess for the elliptic grid generation method and accelerates convergence time for the elliptic method. Another advantage of using the TFI method as an initial guess is that it prevents the grids generated by the elliptic (O-type) method from folding. The formulations for the TFI and elliptic grid generation techniques were investigated in Chapter 2.

6.4 Flow equation solution

If V_∞ and α are known, then from Equations (6.12) and (6.13) one can obtain the stream function ψ at any point in the physical domain as below

$$\psi_2 = \psi_1 + (y_2 - y_1)V_\infty \cos\alpha \tag{6.22}$$

$$\psi_2 = \psi_1 - (x_2 - x_1)V_\infty \sin\alpha \tag{6.23}$$

where subscripts 1 and 2 refer to any two arbitrary grid points in the physical domain. Equations (6.22) and (6.23) are applied to the vertical and horizontal boundaries of the physical domain, respectively. By knowing the values of the stream function ψ on the boundaries of the physical domain as well as on the airfoil surface, we can obtain the values of ψ over the physical domain by applying the Kutta condition as well as using the following formula (by mapping the physical domain onto the computational domain) [the relationships were given in Chapter 2 to find the temperature (Equations (2.40) and (2.41))]

$$\alpha\psi_{\xi\xi} - 2\beta\psi_{\xi\eta} + \gamma\psi_{\eta\eta} = -J^2(P\left(\xi,\eta\right)\psi_\xi + Q\left(\xi,\eta\right)\psi_\eta) \tag{6.24}$$

where

$$\alpha = x_\eta^{\,2} + y_\eta^{\,2}$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta$$

$$\gamma = x_\xi^{\,2} + y_\xi^{\,2}$$

$$J = x_\xi y_\eta - x_\eta y_\xi \left(\text{Jacobian of transformation}\right) \tag{6.25}$$

Equations (6.24) and (6.25) are discretized using the finite difference method.

Kutta condition: For a given airfoil at a given angle of attack, there are an infinite number of valid theoretical solutions, corresponding to an infinite choice of circulation Γ. Figure 6-5 shows two different flows over the same airfoil at the same angle of attack but with different values of Γ. However, experience shows that there is a single value of lift produced by a given airfoil at a given angle of attack. The German mathematician W. M. Kutta was the first to use this trailing edge condition in a theoretical paper in 1902 [133]. The Kutta condition thus states that: The flow leaves the sharp trailing edge of an airfoil smoothly. To apply the Kutta condition in our calculation, we need to consider two possible configuration of the trailing edge. The trailing edge can have a finite angle or can be cusped (Figure 6-6).

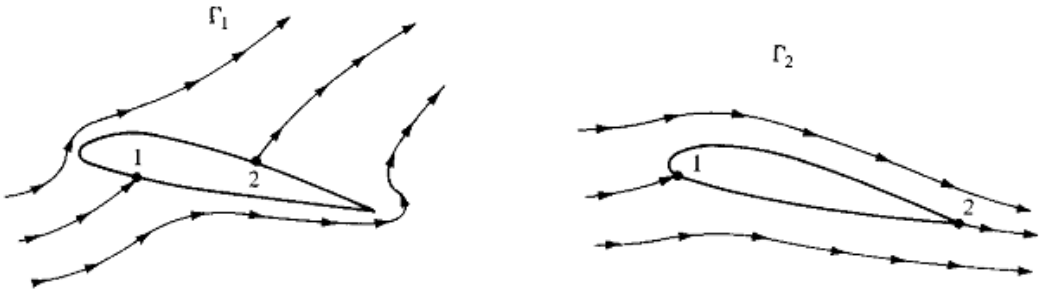
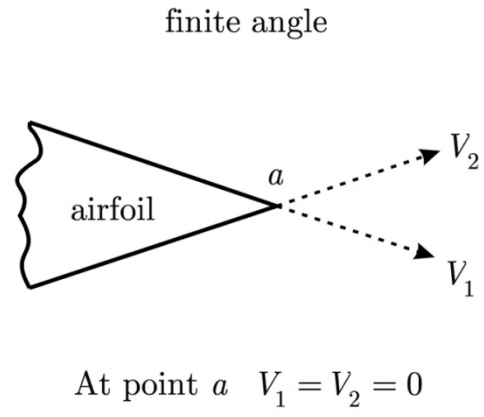
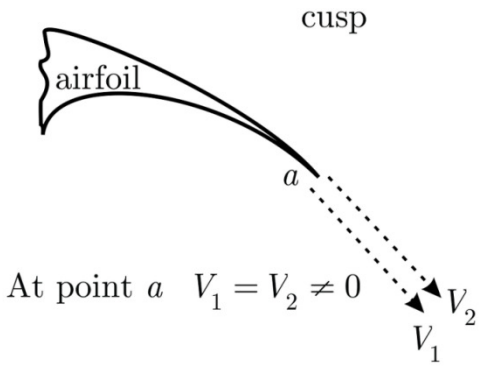


Figure 6-5 Effect of different values of circulation on the potential flow over a given airfoil at a given angle of attack. Points 1 and 2 are stagnation points [132].



a)



b)

Figure 6-6 Different possible shapes of the trailing edge and their relation to the Kutta condition.

Suppose that the velocity along the top surface and bottom surface are V_1 and V_2, respectively. For a finite angle trailing edge, having two finite velocities in two different directions at the same point is physically impossible (Figure 6-6a) and therefore, the only possibility is that both velocities should be zero ($V_1 = V_2 = 0$). For the cusped trailing edge (Figure 6-6b), having two velocities in the same directions at point a shows that both V_1 and V_2 can be finite. However, the pressure at point a is unique and Bernoulli equation states that [132]

$$p_a + \frac{1}{2}\rho V_1^2 = p_a + \frac{1}{2}\rho V_2^2 \tag{6.26}$$

or

$$V_1 = V_2 \tag{6.27}$$

In order to obtain relationships for Kutta condition in terms of the stream function ψ, consider the finite angle trailing edge in the O-type grid scheme shown in Figure 6-7.

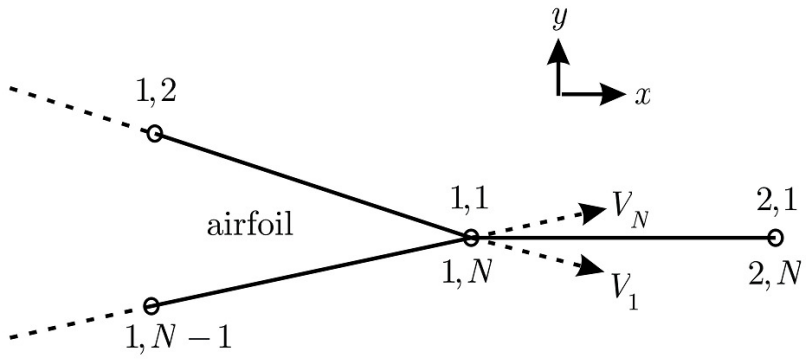


Figure 6-7 Grid notation of the trailing edge.

From Equation (6.7), we have

$$V_x = \psi_y \tag{6.28}$$

From the transformation relationships developed in Chapter 2,

$$\psi_y = \frac{1}{J}\Big[-(x_\eta)(\psi_\xi) + (x_\xi)(\psi_\eta)\Big] \tag{6.29}$$

If V_1 and V_N are the velocities of the grid points (1,1) and $(1,N)$, respectively, the Kutta condition $V_1 = V_2 = 0$ gives

$$V_1 = V_N = 0 \Rightarrow V_{1x} = V_{Nx} = 0$$

$$\frac{1}{J}\Big[-(x_\eta)(\psi_\xi) + (x_\xi)(\psi_\eta)\Big]\Big|_1 = \frac{1}{J}\Big[-(x_\eta)(\psi_\xi) + (x_\xi)(\psi_\eta)\Big]\Big|_N = 0$$

$$-x_\eta\psi_\xi + x_\xi\psi_\eta\Big|_1 = 0 \tag{6.30}$$

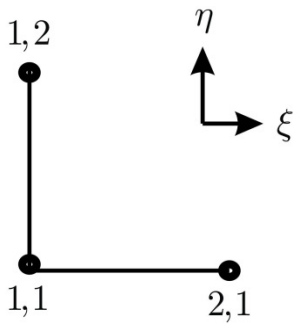


Figure 6-8 Trailing edge grid point (1,1) and its adjacent grid points in the computational domain.

By discretizing Equation (6.30) in the computational domain (Figure 6-8), we get

$$x_\eta\psi_\xi = x_\xi\psi_\eta$$

$$\Big[(x_{1,2} - x_{1,1})\Big]\Big[(\psi_{2,1} - \psi_{1,1})\Big] = \Big[(x_{2,1} - x_{1,1})\Big]\Big[(\psi_{1,2} - \psi_{1,1})\Big]$$

$$\psi_{1,1} = \frac{\psi_{2,1}(x_{1,2} - x_{1,1}) - \psi_{1,2}(x_{2,1} - x_{1,1})}{x_{1,2} - x_{2,1}} \tag{6.31}$$

By considering the wall boundary condition ($\psi_{1,1} = \psi_{1,2}$), we can simplify Equation (6.31) to get

$$\psi_{1,1} = \psi_{2,1} \tag{6.32}$$

Since the grid points (1,1) and $(1, N)$ are the same points in the physical domain, we have

$$\psi_{1,1} = \psi_{1,N} = \psi_{2,1} \tag{6.33}$$

This value is constant on the airfoil surface due to the wall boundary condition.

The derivation of an equation for the cusped trailing edge is more complicated. Consider the cusped trailing edge and the associated grid notation shown in Figure 6-9.

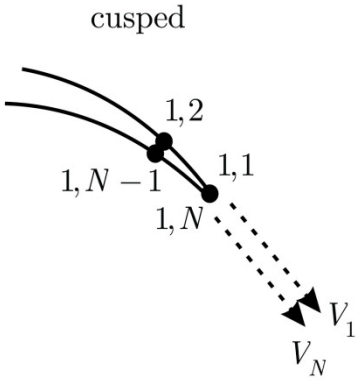


Figure 6-9 Cusped trailing edge and the associated grid notation.

From the Kutta condition for the cusped trailing edge ($V_1 = V_N$) we can write

$$V_1 = V_N \Rightarrow V_{1x} = V_{Nx}$$

Because both vectors V_1 and V_N are equal in the magnitude and direction.

$$\left[\frac{1}{J}(-x_{\eta}\psi_{\xi} + x_{\xi}\psi_{\eta})\right]_1 = \left[\frac{1}{J}(-x_{\eta}\psi_{\xi} + x_{\xi}\psi_{\eta})\right]_N \tag{6.34}$$

But

$$x_{\xi}\Big|_1 = x_{2,1} - x_{1,1}, x_{\xi}\Big|_N = x_{2,N} - x_{1,N} \tag{6.35}$$

Since $x_{2,N} = x_{2,1}$ and $x_{1,N} = x_{1,1}$ we have

$$x_{\xi}\Big|_1 = x_{\xi}\Big|_N \tag{6.36}$$

In similar approach, we have

$$y_{\xi}\Big|_1 = y_{\xi}\Big|_N \tag{6.37}$$

Furthermore, $\psi_{\xi}\Big|_1 = \psi_{2,1} - \psi_{1,1}$ and $\psi_{\xi}\Big|_N = \psi_{2,N} - \psi_{1,N}$. Since $\psi_{1,1} = \psi_{1,N}$ and $\psi_{2,1} = \psi_{2,N}$

$$\psi_{\xi}\Big|_1 = \psi_{\xi}\Big|_N \tag{6.38}$$

Moreover, $\psi_{\eta}\Big|_1 = \psi_{1,2} - \psi_{1,1}$ and $\psi_{\eta}\Big|_N = \psi_{1,N} - \psi_{1,N-1}$.

Since $\psi_{1,1} = \psi_{1,2} = \psi_{1,N-1} = \psi_{1,N}$ (wall boundary condition), we obtain

$$\psi_{\eta}\Big|_1 = \psi_{\eta}\Big|_N = 0 \tag{6.39}$$

By substituting Equations (6.36) through (6.39) into Equation (6.34), we have

$$\frac{1}{(x_{2,1} - x_{1,1})(y_{1,2} - y_{1,1}) - (x_{1,2} - x_{1,1})(y_{2,1} - y_{1,1})} \times$$

$$\Big[-(x_{1,2} - x_{1,1})(\psi_{2,1} - \psi_{1,1}) + 0\Big] =$$

$$\frac{1}{(x_{2,1}-x_{1,1})(y_{1,N}-y_{1,N-1})-(x_{1,N}-x_{1,N-1})(y_{2,1}-y_{1,1})}\times$$

$$\Big[-(x_{1,N}-x_{1,N-1})(\psi_{2,1}-\psi_{1,1})+0\Big] \tag{6.40}$$

By solving Equation (6.40) for $\psi_{1,1}$ (using software Maple), we get

$$\psi_{1,1}=\psi_{2,1} \tag{6.41}$$

In addition, $\psi_{1,N}=\psi_{1,1}=\psi_{2,1}$. Equation (6.41) is the required expression for the cusped trailing angle.

Figure 6-10 and Figure 6-11 show the stream function ψ for both the finite angle (NACA 0012 airfoil with angle of attack of $\alpha=40^{\circ}$ and a free stream velocity of $V_{\infty}=70\frac{\mathrm{m}}{\mathrm{s}}$) and the cusped (NACA 64012 with angle of attack of $\alpha=40^{\circ}$ and a free stream velocity of $V_{\infty}=70\frac{\mathrm{m}}{\mathrm{s}}$) trailing edge, respectively.

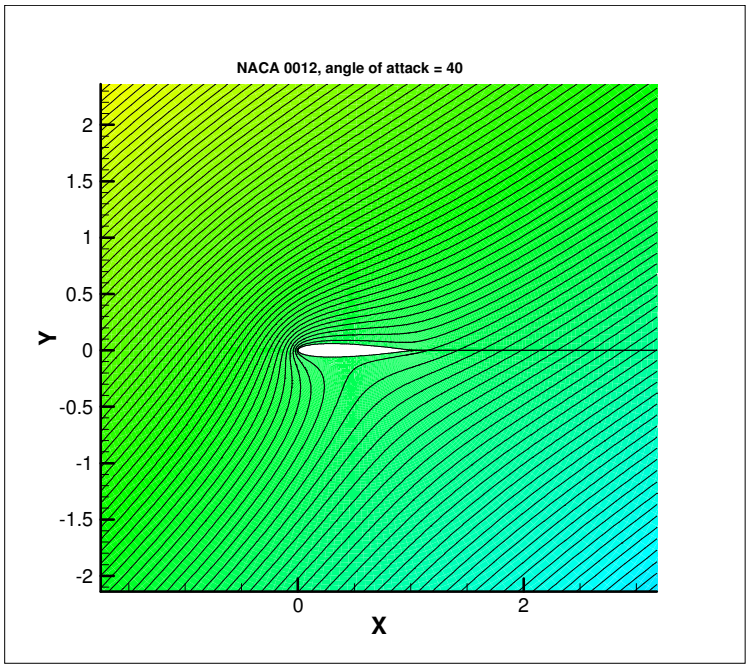


Figure 6-10 Stream function for a finite angle trailing edge. The Figure shows the Kutta condition at the trailing edge.

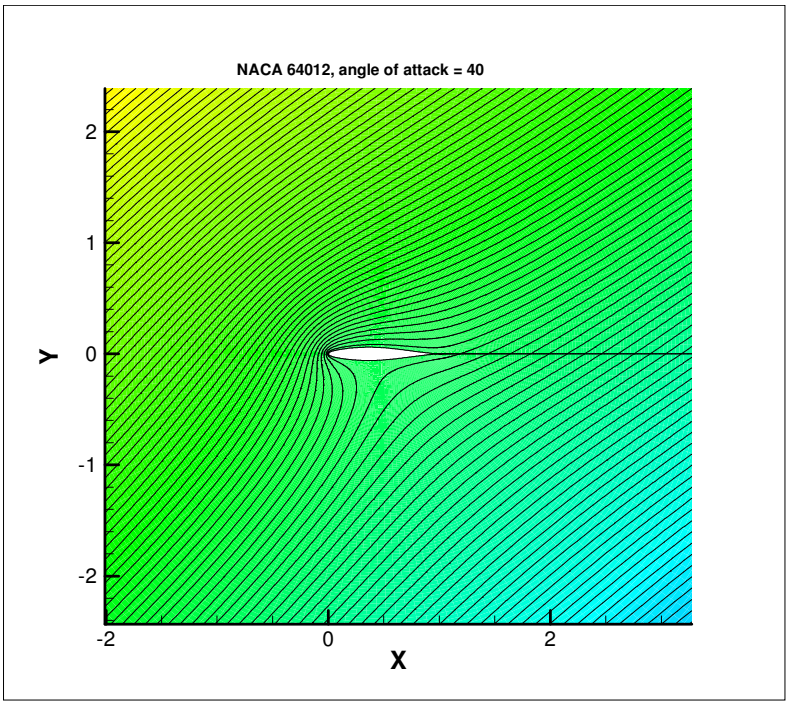


Figure 6-11 Stream function for a cusped trailing edge. The Figure shows the Kutta condition at the trailing edge.

Velocity calculation: There are three sections where the velocity must be known:

1) The outer boundaries (four sides CD, DE, EF, FC of the rectangle shown in Figure 6-2).
2) The airfoil surface (AH in Figure 6-2).
3) The inside of the physical domain.

The velocity values on the outer boundaries are known from the conditions at infinity (using Equations (6.12) and (6.13)). In other words, $x-$ component of the velocity vector (u) on all the outer boundaries is equal to $V_\infty \cos\alpha$ and $y-$ component of the velocity vector (v) on all the outer boundaries is equal to $V_\infty \sin\alpha$. For the inside the physical domain and the airfoil surface, we can use the relationships similar to those for calculation of the temperature obtained in Chapter 2 (Equations (2.11) and (2.12)) which will be

$$u_{i,j} = \left.\frac{\partial \psi}{\partial y}\right|_{i,j} = \frac{1}{J}\left[-(x_\eta)_{i,j}(\psi_\xi)_{i,j} + (x_\xi)_{i,j}(\psi_\eta)_{i,j}\right] \tag{6.42}$$

$$v_{i,j} = -\left.\frac{\partial \psi}{\partial x}\right|_{i,j} = -\frac{1}{J}\left[(y_\eta)_{i,j}(\psi_\xi)_{i,j} - (y_\xi)_{i,j}(\psi_\eta)_{i,j}\right] \tag{6.43}$$

The central and forward difference schemes are used for the inside of the physical domain and the airfoil surface, respectively. After obtaining the components of the velocity vector, the total velocity can be computed by

$$V_{i,j} = \sqrt{u_{i,j}^2 + v_{i,j}^2} \tag{6.44}$$

As stated before, for an incompressible flow, the pressure coefficient can be expressed in terms of velocity only. Thus Equation (6.20) can be used to determine the pressure at any grid point in the domain. Therefore,

$$p_{i,j} = \frac{1}{2}\rho(V_\infty^2 - V_{i,j}^2) + p_\infty \tag{6.45}$$

Validation of the results for the pressure distribution: The results obtained here are compared with the results from using the panel method. The results are obtained by a Fortran compiler (PGI) and computations are run on a PC with Intel Pentium Dual 1.73 and 1 G RAM. The tolerance used in the iterative loops (the mesh generation and the stream function) is 10^{-8}.

Trailing edge with Finite angle:

Valiation case 1: The pressure coefficient distribution (C_p) over a NACA 0012 airfoil at an angle of attack of $\alpha = 9^\circ$ is plotted. The results are compared with the results from [132]. The O-type grid size used in the computation is 155×155. The computation time is 53 seconds.

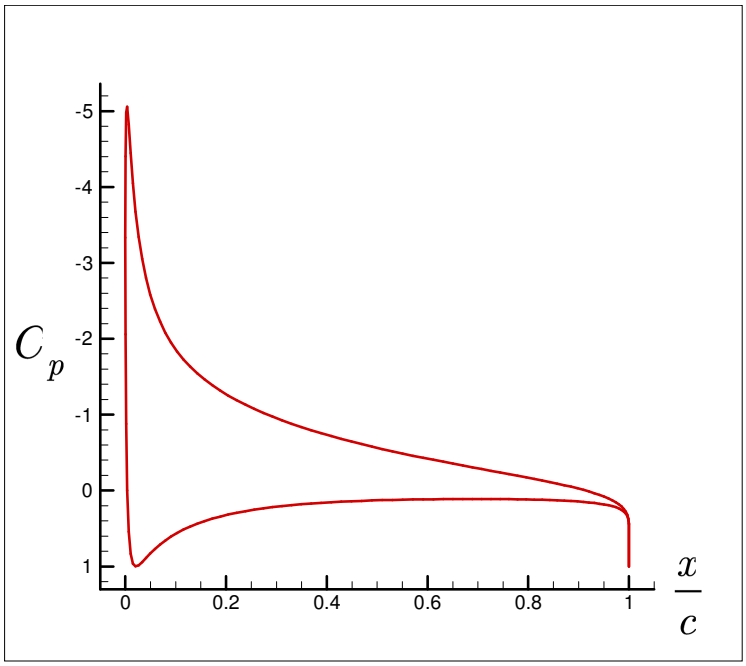


Figure 6-12 Pressure coefficient distribution over a NACA 0012 airfoil at a 9° angle of attack.

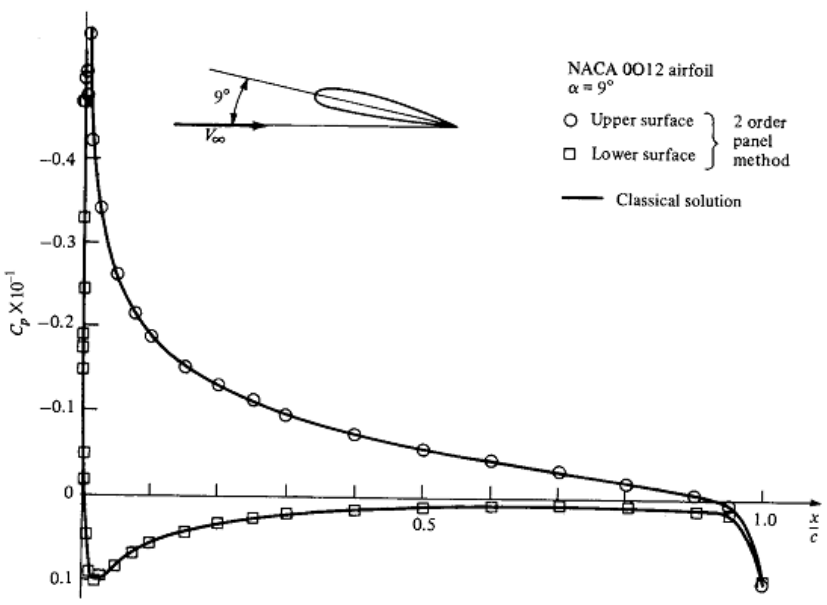


Figure 6-13 Pressure coefficient distribution over a NACA 0012 airfoil; comparison between second order vortex panel method and theoretical results [132].

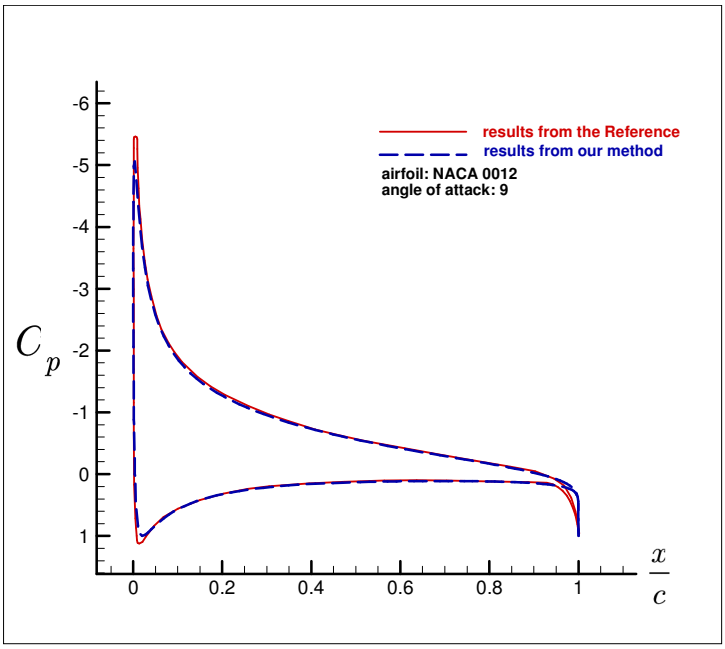


Figure 6-14 Comparison between the results from [132] and the results from our method for validation case 1. The Figure shows an excellent agreement between the results.

Validation case 2: The pressure coefficient distribution (C_p) over a NACA 0024 airfoil at an angle of attack of $\alpha = 0^\circ$ is plotted. The results are compared with the results from [134]. The O-type grid size used in the computation is 155×155. The computation time is 41 seconds.

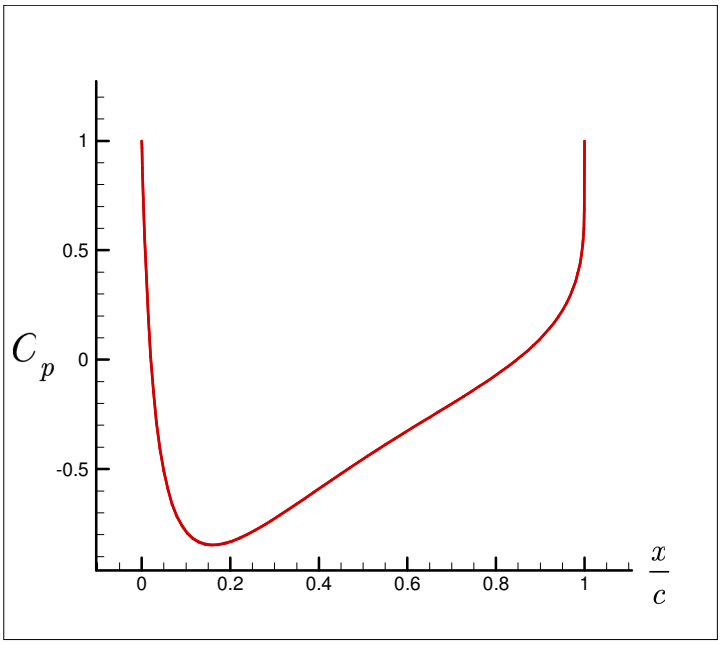


Figure 6-15 Pressure coefficient distribution over a NACA 0024 airfoil at a 0° angle of attack.

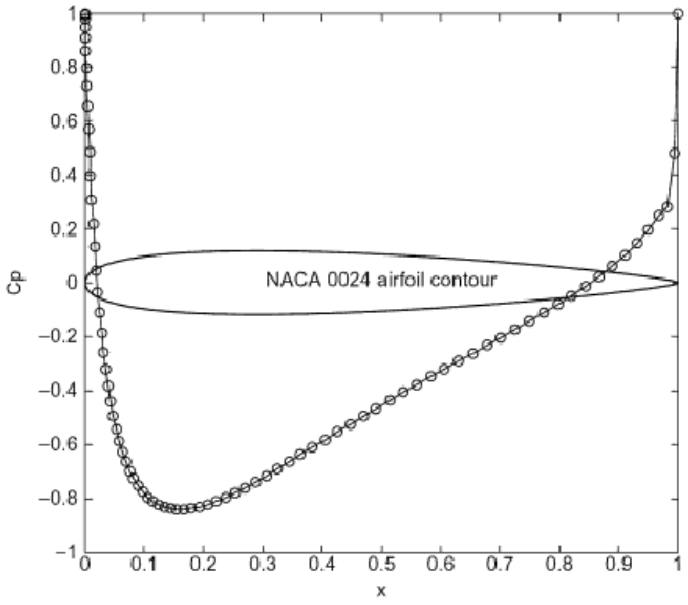


Figure 6-16 Pressure coefficient distribution over a NACA 0024 airfoil at a 0° angle of attack [134].

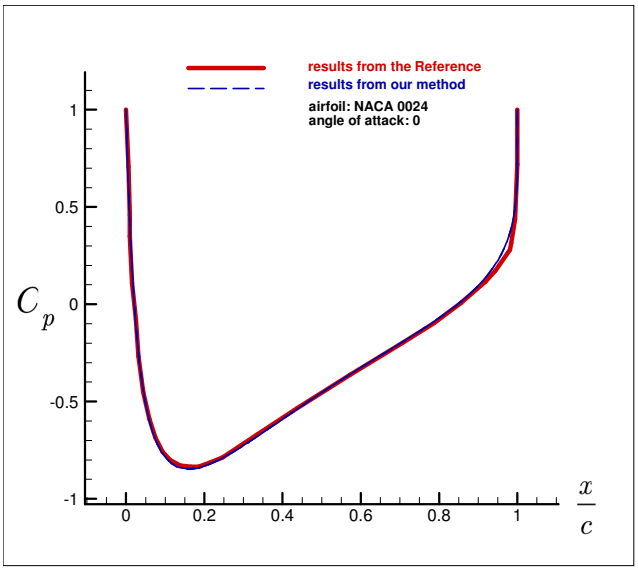


Figure 6-17 Comparison between the results from [134] and the results from our method for validation case 2. The Figure shows an excellent agreement between the results.

Validation case 3: The pressure coefficient distribution (C_p) over a NACA 4414 airfoil at an angle of attack of $\alpha = 2^\circ$ is plotted. The results are compared with the results from the software Xfoil [135]. The O-type grid size used in the computation is 155×155. The computation time is 51 seconds.

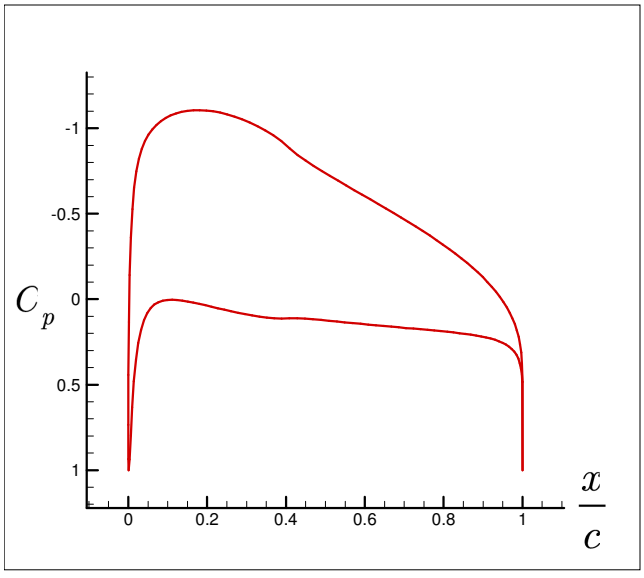


Figure 6-18 Pressure coefficient distribution over a NACA 4414 airfoil at a 2° angle of attack.

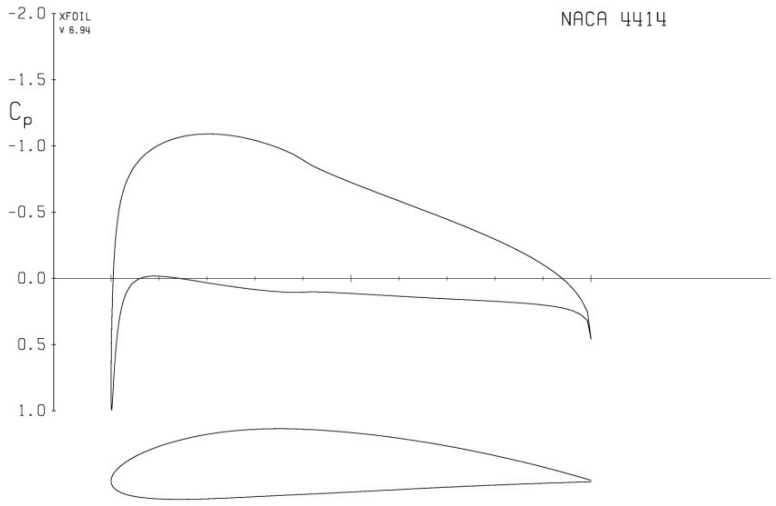


Figure 6-19 Pressure coefficient distribution over a NACA 4414 airfoil at a 2° angle of attack obtained by XFoil.

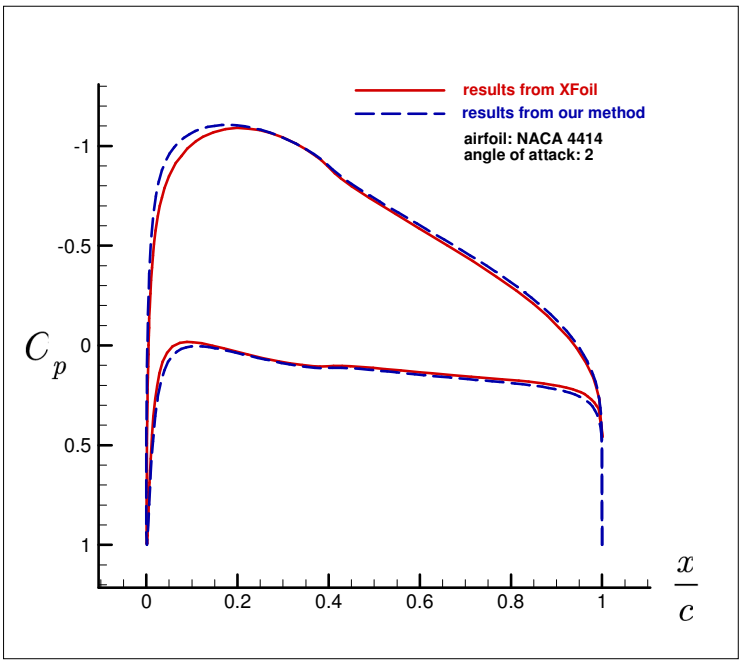


Figure 6-20 Comparison between the results from [135] and the results from our method for validation case 3. The Figure shows an excellent agreement between the results.

Validation case 4: The pressure coefficient distribution (C_p) over a NACA 4412 airfoil at an angle of attack of $\alpha = 10^\circ$ is plotted. The results are compared with the results in [136]. The O-type grid size used in the computation is 155×155. The computation time is 55 seconds.

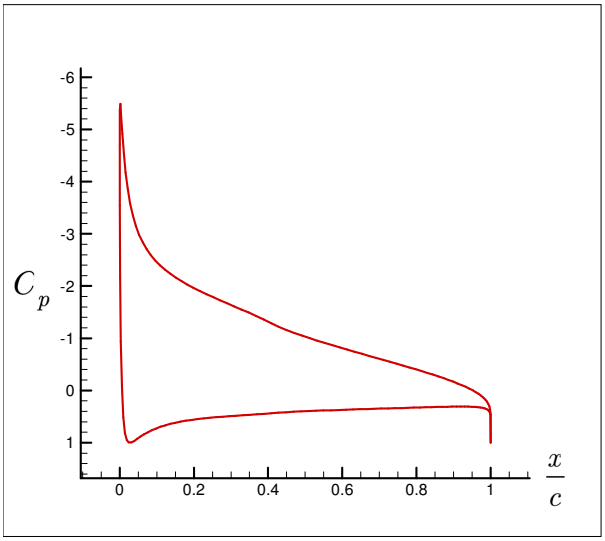


Figure 6-21 Pressure coefficient distribution over a NACA 4412 airfoil at a 10° angle of attack.

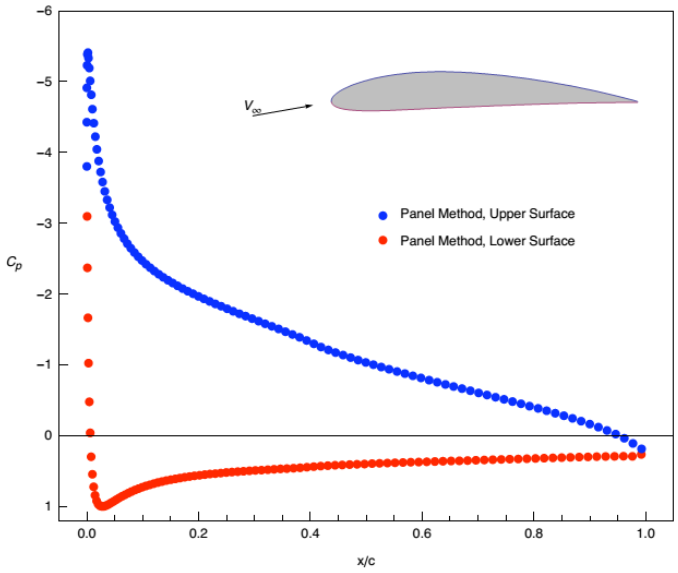


Figure 6-22 Pressure coefficient distribution over a NACA 4412 airfoil at a 10° angle of attack [136].

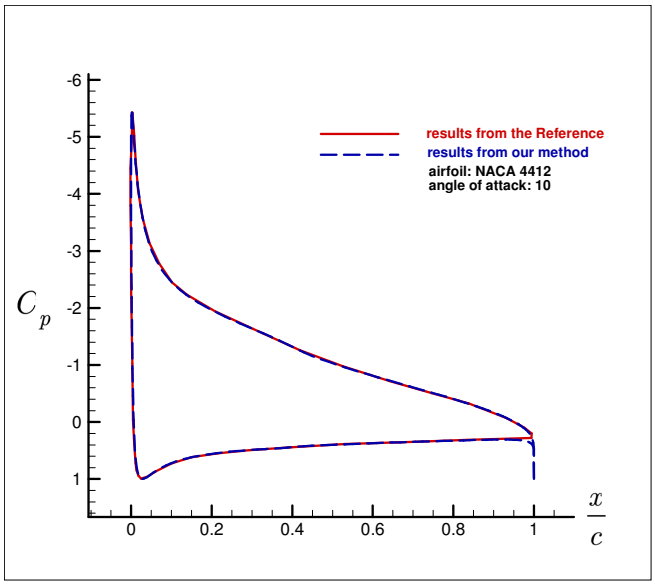


Figure 6-23 Comparison between the results from [136] and the results from our method for validation case 4. The Figure shows an excellent agreement between the results.

Cusped trailing edge:

Validation case 1: The pressure coefficient distribution (C_p) over a NACA 64012 airfoil at an angle of attack of $\alpha = 6^\circ$ is plotted. The results are compared with the results from the software XFLR5 [137]. The O-type grid size used in the computation is 245×245. The computation time is 4 minutes and 15 seconds.

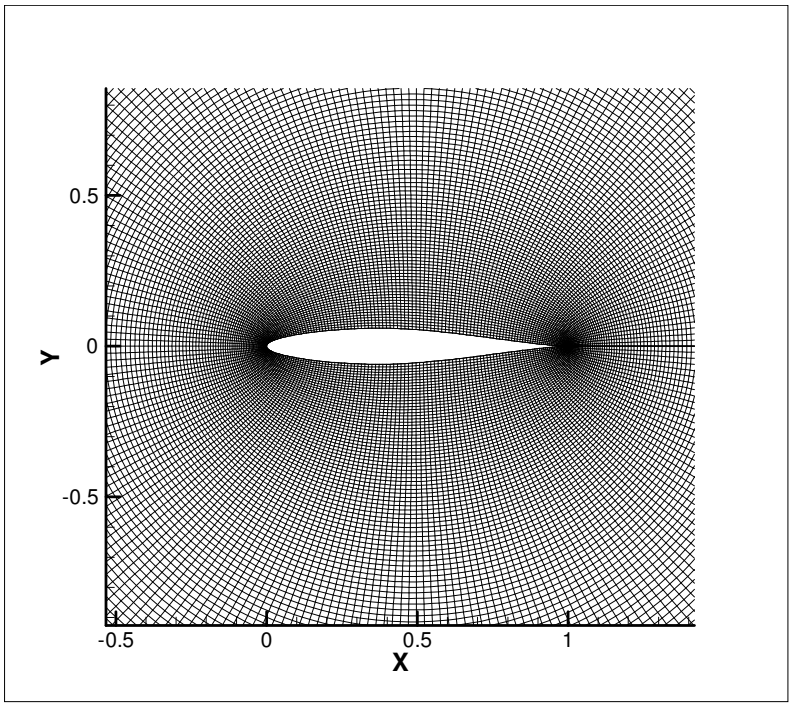


Figure 6-24 O-type grid (elliptic) around the airfoil (NACA 64012) used in validation case 1 (cusped trailing edge).

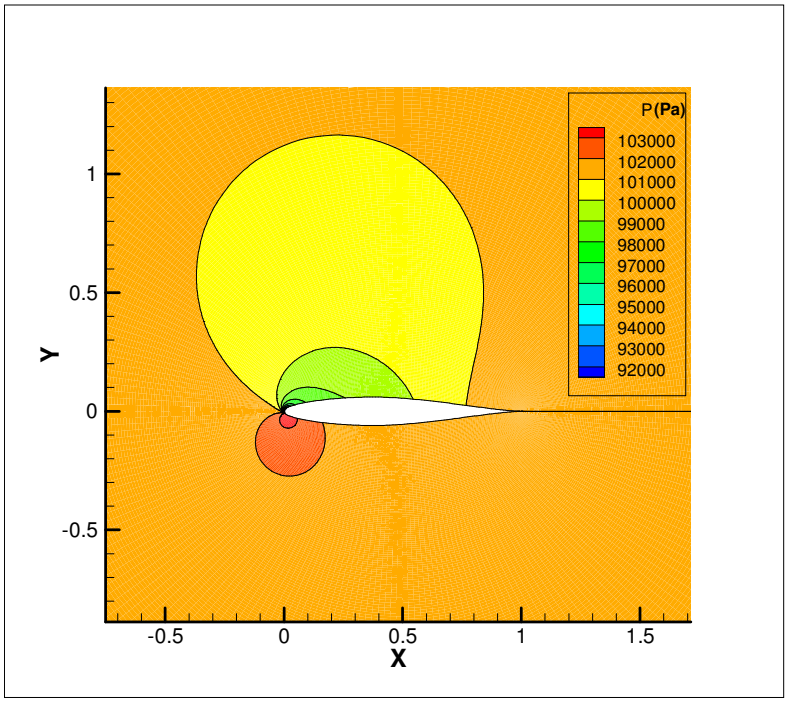


Figure 6-25 Pressure distribution over the airfoil (NACA 64012) used in validation case 1 (cusped trailing edge).

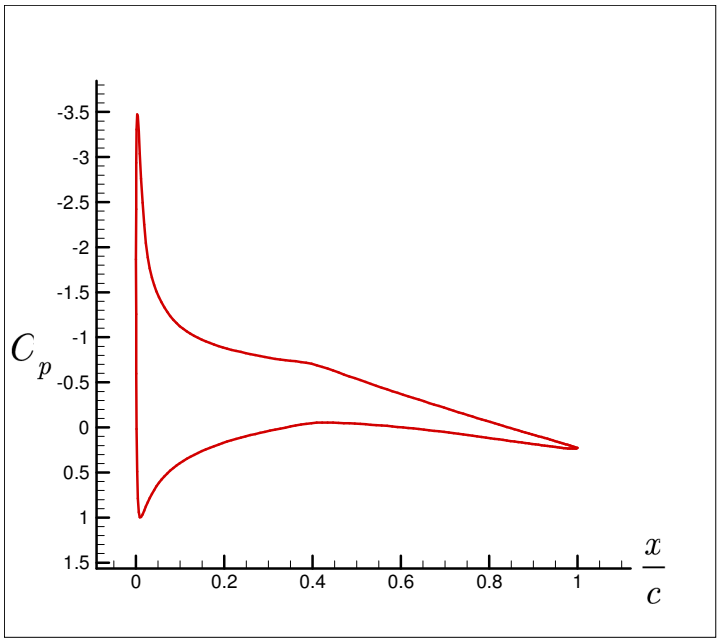


Figure 6-26 Pressure coefficient distribution over a NACA 64012 airfoil at a 6° angle of attack.

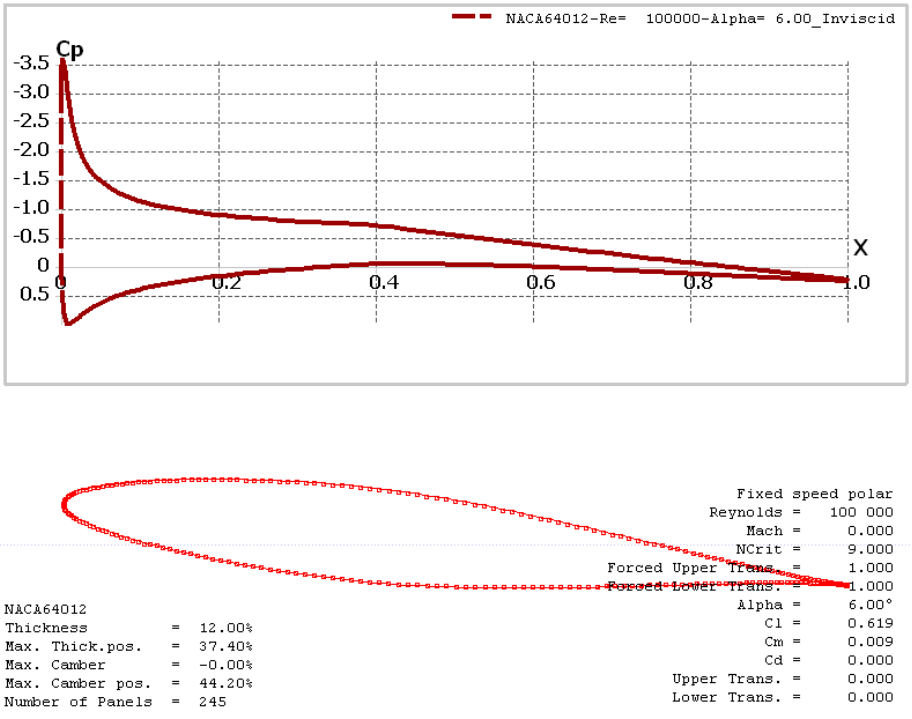


Figure 6-27 Pressure coefficient distribution over a NACA 64012 airfoil at a 6° angle of attack obtained by XFLR5.

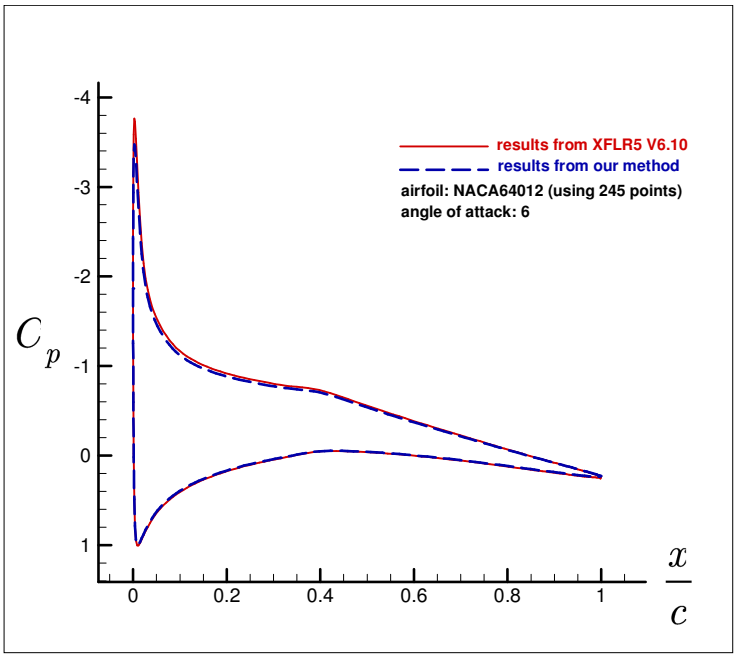


Figure 6-28 Comparison between the results from the software XFLR5 and the results from our method for validation case 1 (cusped trailing edge). The Figure shows an excellent agreement between the results.

Excellent agreement can be obtained by comparing the results from our method and the ones from the panel method given in validation cases for both finite-angle and the cusped trailing edges. As shown in the validation cases results, the maximum value for C_p is exactly equal to 1. The pressure coefficient at the Trailing Edge (T.E.) is equal to unity because the velocity is zero at this stagnation point. Accordingly,

$$V_{T.E.} = 0, C_p = 1 - \frac{V_{T.E.}^2}{V_\infty^2} = 1 - 0 = 1$$

For the cusped trailing edge, $V_{T.E.} \neq 0$. Thus the value of C_p is not equal to 1 ($C_P \neq 1$), as shown in Figure 6-26.

To explore the results from the equations given in this chapter, a comprehensive test case will be given. It includes the distributions for the pressure (Figure 6-29), the velocity (Figure 6-31), the pressure coefficient (Figure 6-32), and the stream function (Figure 6-30) over an NACA 0012 at a

2° angle of attack and the free stream velocity $V_\infty = 70\frac{\text{m}}{\text{s}}$ with an O-gird 185×185. In the following figures, the unit of pressure and velocity are Pa and $\frac{\text{m}}{\text{s}}$, respectively. The computation time is 2 minutes and 22 seconds.

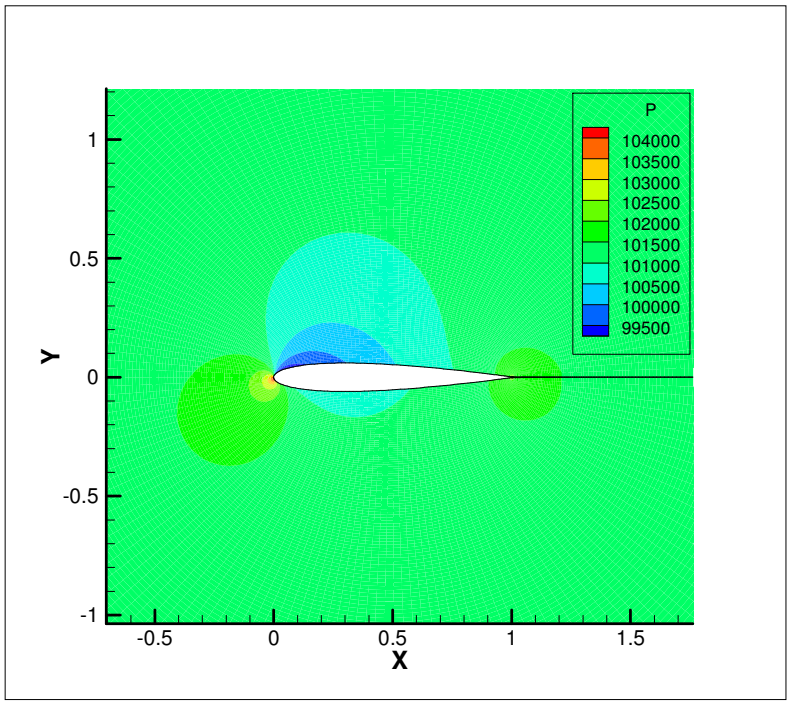


Figure 6-29 Pressure distribution over a NACA 0012 airfoil at a 2° angle of attack.

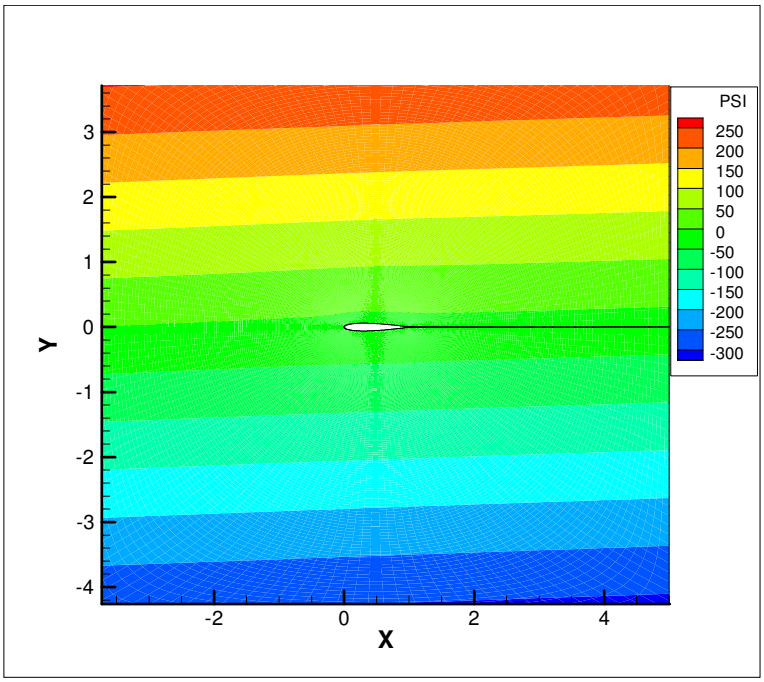


Figure 6-30 Stream function distribution over a NACA 0012 airfoil at a 2° angle of attack.

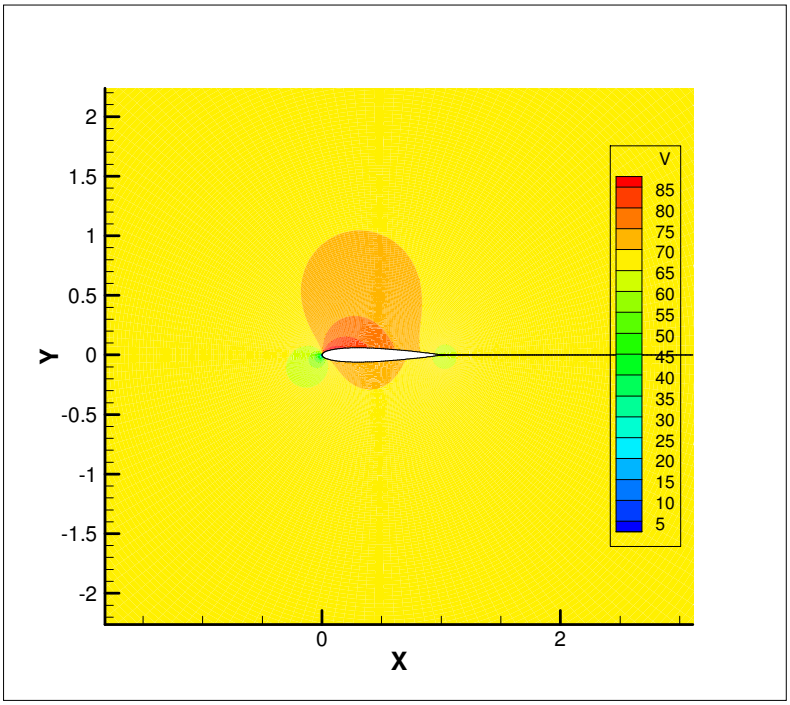


Figure 6-31 Velocity distribution over a NACA 0012 airfoil at a 2° angle of attack.

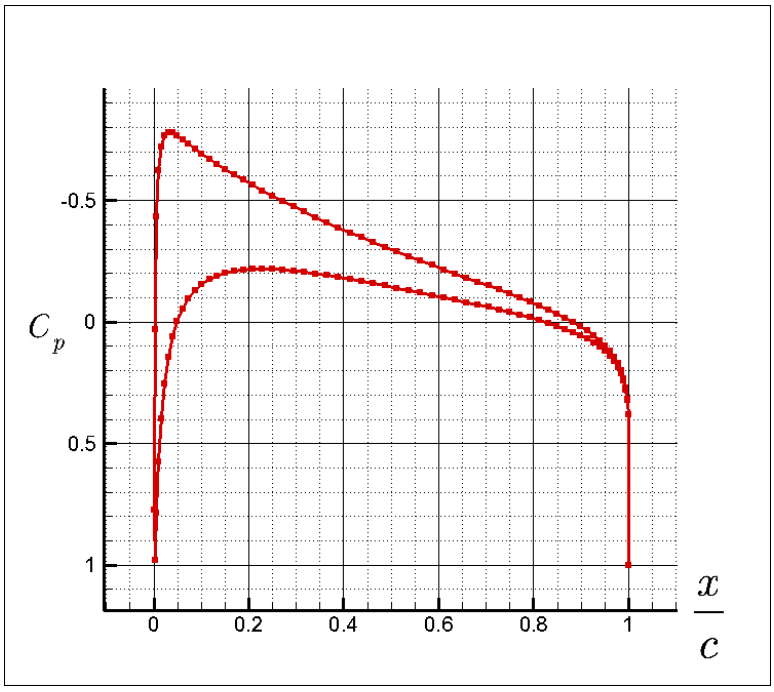


Figure 6-32 Pressure coefficient distribution over a NACA 0012 airfoil surface at a 2° angle of attack.

6.5 Airfoil parameterization

So far, the airfoil surface is parameterized by grid points which result in accurate pressure distribution on the airfoil surface. However, a large number of grid points is required to obtain such accurate results which in turn leads to high computation cost. The design variables are the coordinate (usually $y-$ coordinate) of grid points. Therefore, the optimization process may be inappropriate if there is a large number of design variables since it is difficult to maintain a smooth geometry, the optimization problem will be difficult to solve, and the optimization strategy is likely to fail or be impractical [138]. Thus alternative methods of airfoil surface parameterization are needed. These methods should represent great flexibility in defining the airfoil surface with minimum design variables. Some of the methods include Hicks-Henne functions [104], PARSEC method introduced by Sobieczky [139], and B-splines. In this thesis, in addition to the grid points to represent the airfoil surface, Bezier curves (a special subset of B-spline) are employed due to their

ability to produce airfoil surfaces easily and precisely with only a few control points. In recent years, the use of Bezier curves in the aerodynamic shape optimization has gained much attention [122, 140-155].

Bezier curve: A Bezier curve is a special case of a B-spline curve and is mathematically defined by

$$P(t) = \sum_{i=0}^{n} B_i J_{n,i}(t) \tag{6.46}$$

where

$$J_{n,i}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} \tag{6.47}$$

is Bernstein basis polynomial of degree n. By convention $0^0 \equiv 1$ and $0! \equiv 1$. Here, n, the degree of the Bernstein basis polynomial is one less than the number of points in the Bezier polygon. In other words, number of control points is $n+1$. The points B_i are the vertices of a Bezier polygon or the control points of a Bezier curve. The curve begins at B_0 and ends at B_n. The order of a Bezier curve k is equal to $n+1$. In other words, the order of a Bezier curve is equal to the number of the control points [156]. Figure 6-33 shows a cubic Bezier curve in which the number of control points (B_i) is 4, the degree of the curve (n) is 3, and the order of the curve (k) is 4.

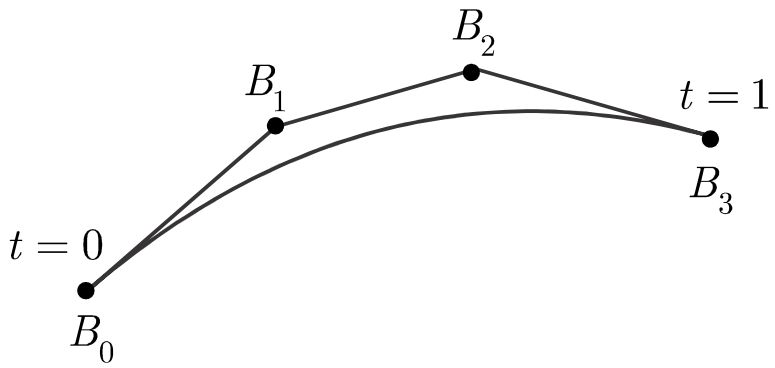


Figure 6-33 A cubic Bezier curve.

In this thesis, two different Bezier curves of order 7 (degree = 6) and of order 11 (degree = 10) will be considered. As will be shown, the Bezier curve of order 7 represents the better optimization performance due to its less design variables. However, this kind of Bezier curve is not able to produce very accurate airfoil shapes. Indeed, it is appropriate to NACA 00XX airfoils only. On the other hand, the Bezier curve of order 11 can successfully generate any airfoil shape with a high degree of accuracy. Therefore, the formulation for the Bezier curve of order 11 only will be given here. The formulation for the Bezier curve of order 7 can be written in a similar fashion.

The parametric Bezier curve of order 11 is as follows

$$n = 10 \Rightarrow \text{number of control points} = 11$$

$$P(t) = \sum_{i=0}^{10} B_i J_{10,i}(t) = B_0 J_{10,0}(t) + \ldots + B_{10} J_{10,10}(t)$$

$$= B_0 \frac{10!}{0!\big(10-0\big)!} t^0 (1-t)^{10-0} + \ldots + B_{10} \frac{10!}{10!\big(10-10\big)!} t^{10} (1-t)^{10-10}$$

Therefore,

$$P(t) = B_0(1-t)^{10} + B_1 10t(1-t)^9 + B_2 45t^2(1-t)^8 + B_3 120t^3(1-t)^7$$

$$B_4 210t^4(1-t)^6 + B_5 252t^5(1-t)^5 + B_6 210t^6(1-t)^4 + B_7 120t^7(1-t)^3$$

$$+B_8 45t^8(1-t)^2 + B_9 10t^9(1-t)^1 + B_{10}t^{10} \tag{6.48}$$

In order to construct the airfoil surface, two Bezier curve will be considered corresponding to the upper and lower surfaces, respectively. Here there are 11 control points (vertices) for each surface. Since the coordinates of the airfoil surface are known, the problem is to determine values for the control points $B_i (i = 0,\ldots,10)$. In other words, our problem is to specify the coordinates of the control points B_i so that the curve passes through the predetermined data points on the airfoil surface. Equation (6.46) can be written in matrix from as follows

$$\big[P(t)\big] = \big[J(t)\big]\big[B\big] \tag{6.49}$$

If the number of the chosen data points on the airfoil surface is m and the degree of Bezier curve is n, then $\left[P(t)\right]$ is a $m\times 2$ matrix, $\left[J(t)\right]$ is a $m\times(n+1)$ matrix, and $\left[B\right]$ is a $(n+1)\times 2$ matrix. Two columns of the matrix $\left[P(t)\right]$ pertain to the $x-$ and $y-$ coordinates of the predetermined data on the airfoil surface. Equation (6.49) can be rewritten as

$$\left[P(t)\right]_{m\times 2} = \left[J(t)\right]_{m\times(n+1)}\left[B\right]_{(n+1)\times 2} \tag{6.50}$$

If $m = n+1$, the matrix $\left[J(t)\right]_{m\times(n+1)}$ will be a square matrix and it can be inverted. In such a case, Equation (6.50) can be written as below to find the matrix $\left\{B\right\}$

$$\left[B\right]_{(n+1)\times 2} = \left[J(t)\right]^{-1}_{m\times(n+1)}\left[P(t)\right]_{m\times 2} \tag{6.51}$$

However, the number of the airfoil surface data points is usually more than the number of control points. In such a case, there are more equations than unknowns and the matrix $\left[J(t)\right]_{m\times(n+1)}$ is no longer a square matrix. Hence it is required to convert it to a square matrix by multiplying both sides of Equation (6.50) by the transpose of $\left[J(t)\right]_{m\times(n+1)}$ as follows

$$\left[J(t)\right]^{\mathrm{T}}_{(n+1)\times m}\left[P(t)\right]_{m\times 2} = \left[J(t)\right]^{\mathrm{T}}_{(n+1)\times m}\left[J(t)\right]_{m\times(n+1)}\left[B\right]_{(n+1)\times 2} \tag{6.52}$$

Thus,

$$\left[B\right]_{(n+1)\times 2} = \left[\left[J(t)\right]^{\mathrm{T}}_{(n+1)\times m}\left[J(t)\right]_{m\times(n+1)}\right]^{-1}\left[J(t)\right]^{\mathrm{T}}_{(n+1)\times m}\left[P(t)\right]_{m\times 2} \tag{6.53}$$

NACA 0015 and TsAGI "B" 12% airfoils produced by Bezier curve with $n=10$ and $m=51$ and their comparison with conventional NACA 0015 and TsAGI "B" 12% airfoils are shown in Figure 6-34 and Figure 6-35, respectively. There is an excellent agreement between two airfoils in each figure.

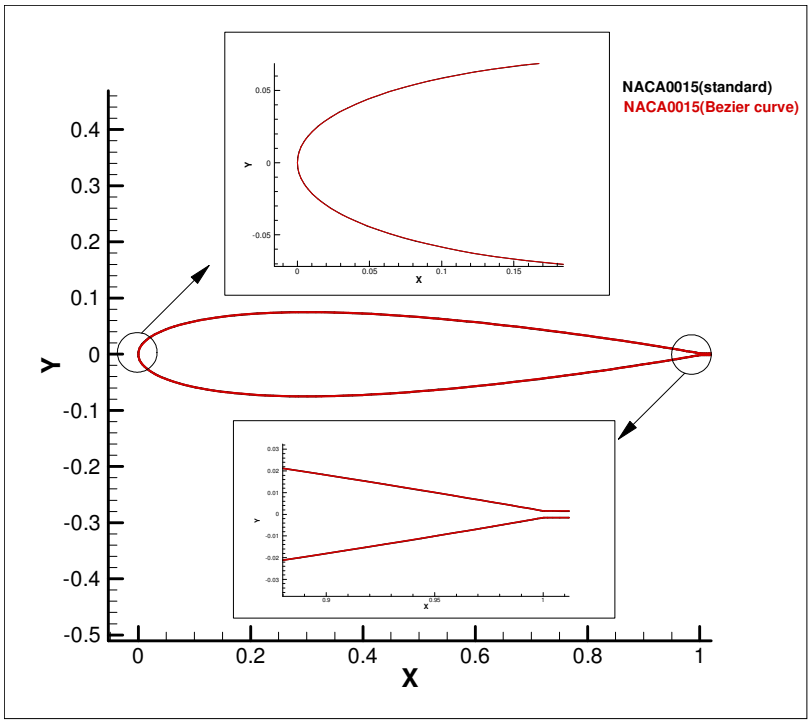


Figure 6-34 Comparison between the standard airfoil and the Bezier curve for a NACA0015.

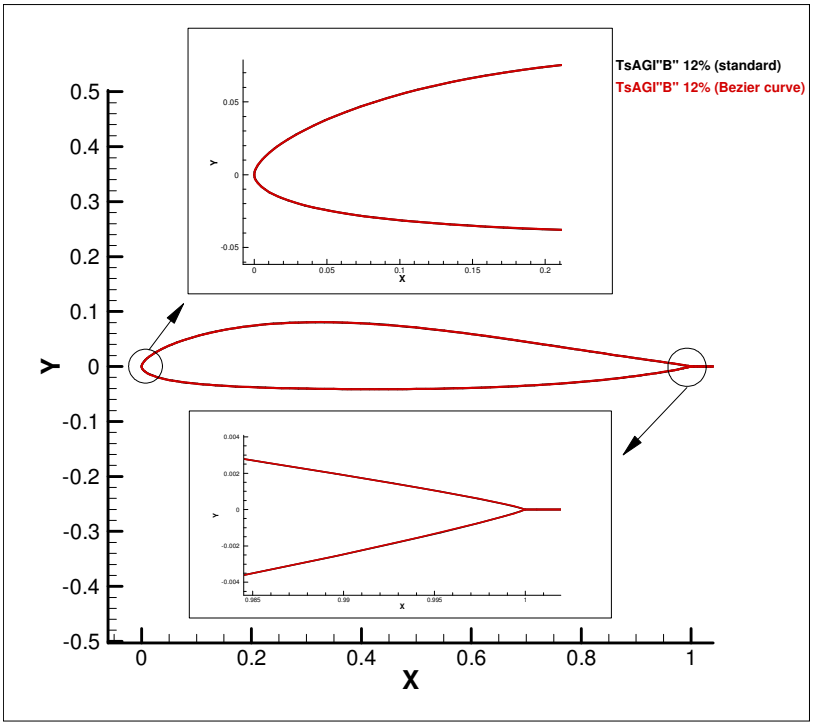


Figure 6-35 Comparison between the standard airfoil and the Bezier curve for a TsAGI “B” 12%.

The predetermined data for the NACA airfoils can be extracted from, for example, the software JavaFoil [157] which is based on the analytical NACA formulations.

NACA 00xx symmetric airfoils: Since the maximum thickness of a NACA 00xx symmetric airfoil will be considered as a design variable, the equation for generating such airfoils is given as follows

$$\pm y_t = \frac{t}{0.2}\Big[0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1015x^4\Big] \quad (6.54)$$

where x is coordinates along the chord of the airfoil, from 0 to c (c is the *chord length* and is assumed equal to 1), y_t is the thickness coordinates above and below the line extending along the length of the airfoil, and t is maximum thickness of the airfoil in percentage of chord (i.e. t in a 15% thick airfoil would be 0.15). Equation (6.54) can be used to find the $y-$ coordinates of a NACA 00xx symmetric airfoil by knowing the values for x and t. As will be shown, the maximum thickness of such airfoils will also be considered as a design variable. By optimizing the thickness, the optimal shape for such airfoils will be obtained. This kind of optimization problem, however, is not comprehensive and produces the optimal NACA 00xx symmetric airfoils only. In summary, three kinds of design variable will be considered in this thesis for airfoil shape optimization which are grid points on a given airfoil surface extracted from, say, the software JavaFoil, the Bezier curve control points, and the maximum thickness of NACA 00xx symmetric airfoils.

6.6 Shape optimization

Several objective functions may be considered for the aerodynamic shape optimization problems including maximizing the lift-drag ratio, maximizing the lift, and minimizing the drag. In the framework of this thesis, the shape optimization problem will be to infer the shape an airfoil should have so that the pressure distribution on the airfoil surface matches a prescribed one (an inverse problem). In inverse design problem, the desired pressure distribution of the target design may be specified a priori.

***Design Variable* (*DV*):** Here the airfoil grid points, the Bezier curve control points, and the maximum thickness of NACA00xx airfoils are considered as design variables. Therefore,

Case (1): The airfoil grid points as design variable

Case (2): The Bezier curve control points as design variable

Case (3): The maximum thickness of NACA00xx airfoils

Case (1): The mathematical expression for the objective function considered for Case (1) can be stated as

$$\mathcal{J} = \sum_{j=2, j\neq\frac{N+1}{2}}^{N-1} (P_{1,j} - P_{d(1,j)})^2 \tag{6.55}$$

where $P_{(1,j)}$ is the pressure at grid points $F_{1,j}$ on the airfoil surface and $P_{d(1,j)}$ is the desirable pressure at grid points $F_{1,j}$ on the airfoil surface (Figure 6-36). The aim is to minimize $\mathcal{J}$ and to reach the desirable pressure distribution by changing the position of the grid points on the airfoil surface. Since the $x-$ coordinates of the grid points can be constant during the optimization process, only the $y-$ coordinates of the grid points are considered as design variables. Two end points of airfoil, namely, *leading edge* $(j = \frac{N+1}{2})$ and *trailing edge* $(j = 1, N)$ are fixed. Thus they are not considered as design variables.

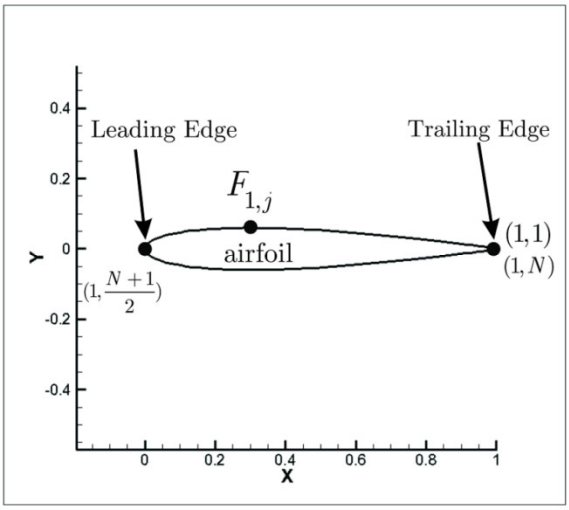


Figure 6-36 Illustration of the airfoil surface points to be optimized so that the objective function reaches a minimum.

Case (2): The mathematical expression for the objective function considered for Case (2) can be stated as

$$\mathcal{J} = \sum_{i=1}^{2m-4} (P_{iB} - P_{iB_d})^2 \tag{6.56}$$

where m is the number of the predetermined data on each of the upper and the lower surfaces of the airfoil, P_{iB} is the pressure at point i of the airfoil surface generated by the Bezier curve, and P_{iB_d} is the desirable pressure at point i . Why $2m-4$? m data points for the upper surface, m data points for the lower surface, and the leading and the trailing edges for two surfaces are considered fixed. The aim is to minimize $\mathcal{J}$ and to reach the desirable pressure distribution by changing the $y-$position of the control points $B_i (i = 1,\ldots,9)$ on each of the upper and the lower surfaces of the airfoil (see Figure 6-37). B_0 and B_{10}, which are concerned with the leading edge and the trailing edge, respectively, are considered fixed for both upper and lower surfaces. Therefore, for the shape optimization problem with a Bezier curve of order 11, we have $2\times(11-2)=18$ design variables. For the shape optimization problem with a Bezier curve of order 7, we have $2\times(7-2)=10$

design variables. The reason for considering these two kinds of the Bezier curve is twofold:

1) To show that the optimization problem will be more successful if we have less number of design variable.
2) To have a very accurate and flexible representation of the airfoil shapes, a degree of at least 10 should be used.

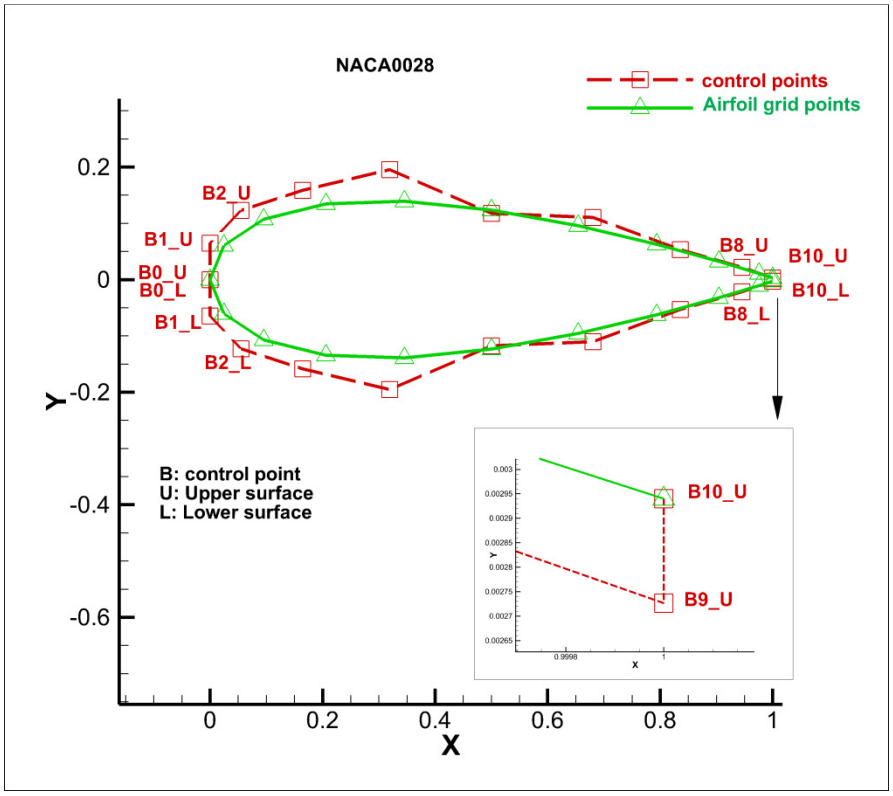


Figure 6-37 Illustration of the Bezier control points (B_i) to be optimized so that the objective function (Equation (6.56)) reaches a minimum.

Case (3): The airfoil surface is generated by the analytical NACA formula (Equation(6.54)) and the maximum thickness is considered as the design variable. To show the accuracy of the sensitivity scheme, the upper and lower airfoil surfaces are generated separately and hence the design variables will be two maximum thicknesses in the upper and lower airfoil surfaces. As shown in

Figure 6-38, if the indices 1 and 2 denote the location of maximum thickness on the upper and lower airfoil surfaces, respectively, then the mathematical expression for the objective function considered for Case (3) is as follows

$$\mathcal{J} = \sum_{i=1}^{2} (P_i - P_{d(i)})^2 \tag{6.57}$$

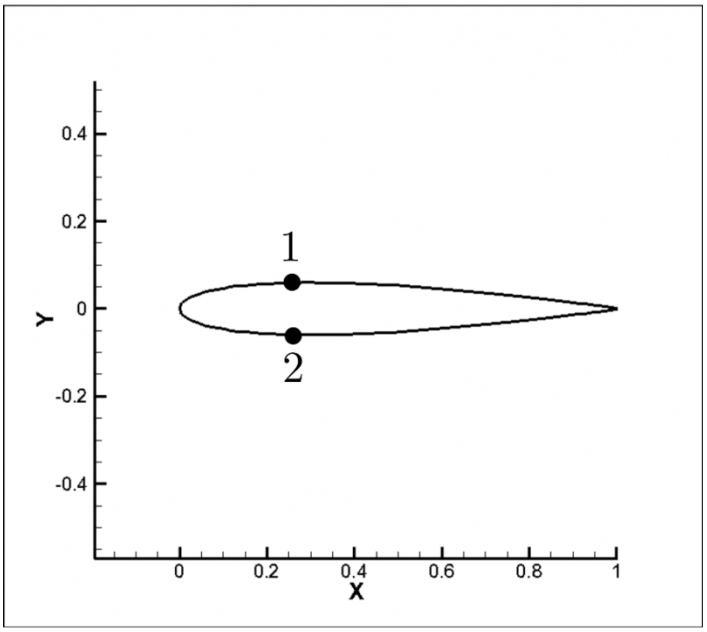


Figure 6-38 The location for the maximum thickness on the upper and lower airfoil surfaces. $y-$ coordinates of the points 1 and 2 are considered as the design variables.

6.7 Sensitivity analysis

Airfoil grid points as design variables:

As explained in preceding chapters, the sensitivity analysis can be carried out by perturbations methods. Suppose we wish to calculate the sensitivity of pressure of nodes on the airfoil surface (see Figure 6-36), $P_{1,j}$

$(j = 2,\ldots,N-1, j \neq \frac{N+1}{2})$, to the $y-$position of the nodes on the airfoil surface, $y_{1,j'}\,(j' = 2,\ldots,N-1, j' \neq \frac{N+1}{2})$. The sensitivity analysis can be performed by introducing small perturbations to the $y-$coordinate of each point on the airfoil surface, individually. The grid generation and flow problem may be solved for this perturbed shape to obtain the new values for the pressure $P_{1,j}$. Using these values for the pressure, the dependency of the pressure $P_{1,j}$ to the perturbation of the $y-$ position of points of coordinates $(1,j')$, $y_{1,j'}$, can be evaluated. The finite difference method may be used to formulate these sensitivities as follows

$$\frac{\partial P_{1,j}}{\partial y_{1,j'}} = \frac{P_{1,j}(y_{1,j'} + \varepsilon y_{1,j'}) - P_{1,j}(y_{1,j'})}{\varepsilon y_{1,j'}} \tag{6.58}$$

where ε may be, say, 10^{-6}. The term $\varepsilon y_{1,j'}$ is the perturbation in the $y-$ position of points of coordinates $(1,j')$, $y_{1,j'}$. Since the sensitivity of each pressure $P_{1,j}\,(j = 2,\ldots,N-1, j \neq \frac{N+1}{2})$ to each $y-$ position of points of coordinates $(1,j')\,(j' = 2,\ldots,N-1, j' \neq \frac{N+1}{2})$ is required, the computation of the sensitivity coefficients using this method requires $(N-3)$ additional solutions of the flow problem. Therefore, this method is only suitable when the number of points on the airfoil surface is small. Thus for the airfoil shape optimization problem, which demand a fine grid to obtain accurate results, the perturbation method using the finite difference method will be of high computation cost. In this chapter, we will expand the novel method used in evaluating the sensitivity matrix in the shape optimization of heat transfer problems. As will be shown, it requires only one solution of the flow problem (at each iteration) to compute all sensitivity coefficients.

With regard to Equation (6.55), the following equation can be written in order to calculate the Jacobian matrix

$$\frac{\partial \boldsymbol{\mathcal{J}}}{\partial y_{1,l}} = 2 \sum_{j=2, j\neq\frac{N+1}{2}}^{N-1} (P_{1,j} - P_{d(1,j)}) \frac{\partial P_{1,j}}{\partial y_{1,l}} \tag{6.59}$$

where $(j = 2,\ldots,N-1, j \neq \frac{N+1}{2})$ and $(l = 2,\ldots,N-1, l \neq \frac{N+1}{2})$. The expression $\frac{\partial P_{1,j}}{\partial y_{1,l}}$ in the above relation is called the Jacobian coefficient. In this case, the sensitivity matrix can be expanded as

$$\mathbf{Ja}_y = \begin{bmatrix} \frac{\partial P_{1,2}}{\partial y_{1,2}} & \frac{\partial P_{1,2}}{\partial y_{1,3}} & \frac{\partial P_{1,2}}{\partial y_{1,4}} & \cdots & \frac{\partial P_{1,2}}{\partial y_{1,N-1}} \\ \frac{\partial P_{1,3}}{\partial y_{1,2}} & \frac{\partial P_{1,3}}{\partial y_{1,3}} & \frac{\partial P_{1,3}}{\partial y_{1,4}} & \cdots & \frac{\partial P_{1,3}}{\partial y_{1,N-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial P_{1,N-1}}{\partial y_{1,2}} & \frac{\partial P_{1,N-1}}{\partial y_{1,3}} & \frac{\partial P_{1,N-1}}{\partial y_{1,4}} & \cdots & \frac{\partial P_{1,N-1}}{\partial y_{1,N-1}} \end{bmatrix} \tag{6.60}$$

Since the physical domain is mapped onto the computational one, the chain rule introduced in Chapter 2 may be used to correlate variables in the two domains. Therefore,

$$\frac{\partial P_{1,j}}{\partial x_{1,l}} = \frac{\partial P_{1,j}}{\partial \xi}\frac{\partial \xi}{\partial x_{1,l}} + \frac{\partial P_{1,j}}{\partial \eta}\frac{\partial \eta}{\partial x_{1,l}} \tag{6.61}$$

$$\frac{\partial P_{1,j}}{\partial y_{1,l}} = \frac{\partial P_{1,j}}{\partial \xi}\frac{\partial \xi}{\partial y_{1,l}} + \frac{\partial P_{1,j}}{\partial \eta}\frac{\partial \eta}{\partial y_{1,l}} \tag{6.62}$$

As pointed out before, the $x-$ coordinate of the grid points are considered fixed and they are not included in the design variables. Thus Equation (6.61) is written here to derive the required relations for the sensitivity coefficients. By interchanging x and ξ, and y and η, and solving the derived equations for $\frac{\partial P}{\partial x}$ and $\frac{\partial P}{\partial y}$, we finally obtain

$$\frac{\partial P_{1,j}}{\partial y_{1,l}} = \frac{1}{J}\Big[-(x_\eta)_{1,l}(P_\xi)_{1,j} + (x_\xi)_{1,l}(P_\eta)_{1,j}\Big] \tag{6.63}$$

where $J = (x_\xi y_\eta - x_\eta y_\xi)_{1,l}$ is the Jacobian of the transformation. Using the finite difference method to discretize the equations in the computational

domain, we can write appropriate algebraic approximations for all partial derivatives involved in the above equation. Therefore,

$$(P_{\xi})_{1,j} = \frac{-3P_{1,j} + 4P_{2,j} - P_{3,j}}{2} \tag{6.64}$$

$$(P_{\eta})_{1,j} = \frac{P_{1,j+1} - P_{1,j-1}}{2} \tag{6.65}$$

$$(x_{\xi})_{1,l} = \frac{-3x_{1,l} + 4x_{2,l} - x_{3,l}}{2} \tag{6.66}$$

$$(x_{\eta})_{1,l} = \frac{x_{1,l+1} - x_{1,l-1}}{2} \tag{6.67}$$

which are based on the central and the forward differences. Equations (6.64) through (6.67) are employed to calculate the sensitivity coefficients in Equation (6.60).

Bezier control points as design variables:

With regard to Equation (6.56) and considering the control points of the Bezier curve as design variable, we can write

$$\frac{\partial \boldsymbol{\mathcal{J}}}{\partial B_{y_l}} = 2\sum_{i=1}^{2m-4}(P_{iB} - P_{iB_d})\frac{\partial P_{iB}}{\partial B_{y_l}} \tag{6.68}$$

Using the chain rule, we can write

$$\frac{\partial P_{iB}}{\partial B_{y_l}} = \frac{\partial P_{iB}}{\partial y_{i'B}}\frac{\partial y_{i'B}}{\partial B_{y_l}} \tag{6.69}$$

where $y_{i'B}(i' = 1,\ldots,2m-4)$ are the $y-$ coordinate of the predetermined grid points to be passed by the Bezier curve and $B_{y_l}(l = 1,\ldots 18)$ are the $y-$ coordinate of Bezier control points whose number is equal to 18 (9 for each of the upper and lower surfaces). The term $\dfrac{\partial P_{iB}}{\partial y_{i'B}}$ can be computed by the

expressions derived for Case (1) [Equation (6.60)]. The size of the matrix formed by the arrays $\frac{\partial P_{iB}}{\partial y_{i'B}}$ is $(2m-4)\times(2m-4)$. The term $\frac{\partial y_{i'B}}{\partial B_{y_l}}$ can be easily evaluated by taking derivative of Equation (6.48) with respect to the control points B (noting that $[P(t)] \equiv [x(t), y(t)]$). The control points may be renumbered so that $B_{y_1} = B_{y_{9U}}$, $B_{y_2} = B_{y_{8U}}$,..., $B_{y_9} = B_{y_{1U}}$ and $B_{y_{10}} = B_{y_{1L}}$, $B_{y_{11}} = B_{y_{2L}}$,..., $B_{y_{18}} = B_{y_{9L}}$. The indices U and L denote the upper and lower surfaces, respectively. The direction of numbering is from right to left for the upper surface and from left to right for the lower surface. The reason for this renumbering is the compatibility with the grid point data reading (most of the airfoil data are in this format) as well as the pressure reading to compute the objective function (Equation(6.56)). However, we should note that the Bezier curve evaluation is from left to right for both the upper and lower surfaces. The size of the matrix formed by the arrays $\frac{\partial y_{i'B}}{\partial B_{y_l}}$ is $(2m-4)\times 18$.

Because the upper and lower surfaces are constructed separately, the variation of y of the upper surface with respect to the change in position of the lower surface control points as well as the variation of y of the lower surface with respect to the change in position of the upper surface control points is zero.

Maximum thickness as design variables:

In a similar derivation to Case (1), the sensitivity matrix for Case (3) will be

$$\mathbf{Ja} = \begin{bmatrix} \frac{\partial P_1}{\partial y_1} & \frac{\partial P_1}{\partial y_2} \\ \frac{\partial P_2}{\partial y_1} & \frac{\partial P_2}{\partial y_2} \end{bmatrix} \tag{6.70}$$

6.8 Optimization method

In this chapter, two powerful optimization methods, namely, the Conjugate Gradient Method and the Quasi-Newton Method will be used. For the airfoil grid points as a design variable (Case (1)) both optimization methods will be employed. However, for the Bezier curve control points as a design variable (Case (2)), only the quasi-Newton method will be used. For Case (3) (the maximum thickness of NACA00xx airfoils as a design variable), only the conjugate gradient method will be employed. The application of the conjugate gradient method in the shape optimization problems in heat transfer was explained in preceding chapters and only the algorithm to obtain the optimal shape for the airfoil will be outlined here.

1. Specify the physical domain, the boundary conditions, the problem conditions such as the free stream velocity and the angle of attack, and the desired airfoil surface pressure distribution.
2. Generate the boundary fitted grids using the grid generation methods described earlier.
3. Solve the direct flow problem of finding the pressure values at any grid points of the physical domain and hence the airfoil surface.
4. Using Equation (6.55), compute the objective function $(\boldsymbol{\mathcal{J}}^{(\mathrm{k})})$.
5. If the value of the objective function obtained in step 4 is less than the specified stopping criterion, the optimization is finished. Otherwise, go to step 6.
6. Compute the sensitivity matrix $(\mathbf{Ja})$ from Equation (6.60).
7. Compute the gradient direction $\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})}$ from Equation (6.59).
8. Compute the conjugation coefficient $\gamma^{(\mathrm{k})}$ from the following equation (the Polak-Ribiere formula)

$$\gamma^{(\mathrm{k})} = \frac{\left[\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})}\right]^{\mathrm{T}}(\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})} - \nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}-1)})}{\| \nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}-1)} \|^2} = \frac{\left[\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})}\right]^{\mathrm{T}}(\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})} - \nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}-1)})}{\left[\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}-1)}\right]^{\mathrm{T}}\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}-1)}} \quad (6.71)$$

   For $\mathrm{k} = 0$, set $\gamma^{(0)} = 0$.
9. Compute the direction of descent $\mathbf{d}^{(\mathrm{k})}$ from the following equation

$$\mathbf{d}^{(\mathrm{k})} = \nabla \boldsymbol{\mathcal{J}}^{(\mathrm{k})} + \gamma^{(\mathrm{k})} \mathbf{d}^{(\mathrm{k}-1)} \tag{6.72}$$

10. Compute the search step size $\beta^{(\mathrm{k})}$ from the following equation

$$\beta^{(\mathrm{k})} = \frac{[\mathbf{Ja}^{(\mathrm{k})} \mathbf{d}^{(\mathrm{k})}]^{\mathrm{T}} [P_{1,j} - P_{d(1,j)}]}{[\mathbf{Ja}^{(\mathrm{k})} \mathbf{d}^{(\mathrm{k})}]^{\mathrm{T}} [\mathbf{Ja}^{(\mathrm{k})} \mathbf{d}^{(\mathrm{k})}]} \tag{6.73}$$

11. Evaluate the new $y-$ coordinates of the airfoil surface grid nodes as follows

$$\mathbf{y}^{(\mathrm{k}+1)} = \mathbf{y}^{(\mathrm{k})} - \beta^{(\mathrm{k})} \mathbf{d}^{(\mathrm{k})} \tag{6.74}$$

12. Set the next iteration $(\mathrm{k} = \mathrm{k}+1)$ and return to step 2.

The above algorithm is for the airfoil grid points as a design variable (Case (1)) only. The algorithm for Case (3) can be expressed in a similar way.

Quasi-Newton Method: Quasi-Newton Method is another powerful optimization method used in this chapter. In quasi-Newton method, the Hessian matrix (which is composed of the second partial derivatives) or inverse of Hessian matrix appeared in Newton method is replaced by an approximation of it. The approximation uses only the first partial derivatives. The *Broydon-Fletcher-Goldfarb-Shanno* (BFGS) method is a quasi-Newton method for solving unconstrained nonlinear optimization which proposed almost simultaneously by Broyden [158], Fletcher [159], Goldfarb [160], and Shanno [161]. In the BFGS method, the Hessian matrix approximation, $\boldsymbol{\mathcal{B}}^{(\mathrm{k})}$, is updated iteratively. The steps of BFGS method can be summarized as follows

1. Specify the physical domain, the boundary conditions, the problem conditions such as the free stream velocity and the angle of attack, and the desired airfoil surface pressure distribution.
2. Generate the boundary fitted grids using the grid generation methods described earlier.

3. Solve the direct flow problem of finding the pressure values at any grid points of the physical domain and hence the airfoil surface.
4. Using Equation (6.55), compute the objective function $(\boldsymbol{\mathcal{J}}^{(\mathrm{k})})$.
5. If value of the objective function obtained in step 4 is less than the specified stopping criterion, the optimization is finished. Otherwise, go to step 6.
6. Compute the sensitivity matrix $(\mathbf{Ja})$ from Equation (6.60).
7. Compute the gradient direction $\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})}$ from Equation (6.59).
8. The initial Hessian matrix approximation, $\boldsymbol{\mathcal{B}}^{(1)}$, is taken as the identity matrix, namely $\boldsymbol{\mathcal{B}}^{(1)} = \mathbf{I}$.
9. Set $\mathbf{S}^{(\mathrm{k})} = -\boldsymbol{\mathcal{B}}^{(\mathrm{k})}\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})}$ and the iteration number as $\mathrm{k} = 1$.
10. Compute the search step size $\beta^{(\mathrm{k})}$ (from Equation (6.73)) in the direction $\mathbf{S}^{(\mathrm{k})}$ and set

$$\mathbf{y}^{(\mathrm{k}+1)} = \mathbf{y}^{(\mathrm{k})} - \beta^{(\mathrm{k})}\mathbf{S}^{(\mathrm{k})} \tag{6.75}$$

11. Repeat the steps 2 to 7 with these new values of $\mathbf{y}$ for the grid points $y-$ coordinates to calculate $\nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}+1)}$.
12. Update the Hessian matrix approximation as

$$\boldsymbol{\mathcal{B}}^{(\mathrm{k}+1)} = \boldsymbol{\mathcal{B}}^{(\mathrm{k})} + (1 + \frac{(\mathbf{g}^{(\mathrm{k})})^{\mathrm{T}}\boldsymbol{\mathcal{B}}^{(\mathrm{k})}\mathbf{g}^{(\mathrm{k})}}{(\mathbf{d}^{(\mathrm{k})})^{\mathrm{T}}\mathbf{g}^{(\mathrm{k})}})\frac{\mathbf{d}^{(\mathrm{k})}(\mathbf{d}^{(\mathrm{k})})^{\mathrm{T}}}{(\mathbf{d}^{(\mathrm{k})})^{\mathrm{T}}\mathbf{g}^{(\mathrm{k})}} - \frac{\mathbf{d}^{(\mathrm{k})}(\mathbf{g}^{(\mathrm{k})})^{\mathrm{T}}\boldsymbol{\mathcal{B}}^{(\mathrm{k})}}{(\mathbf{d}^{(\mathrm{k})})^{\mathrm{T}}\mathbf{g}^{(\mathrm{k})}} - \frac{\boldsymbol{\mathcal{B}}^{(\mathrm{k})}\mathbf{g}^{(\mathrm{k})}(\mathbf{d}^{(\mathrm{k})})^{\mathrm{T}}}{(\mathbf{d}^{(\mathrm{k})})^{\mathrm{T}}\mathbf{g}^{(\mathrm{k})}} \tag{6.76}$$

where

$$\mathbf{d}^{(\mathrm{k})} = \mathbf{y}^{(\mathrm{k}+1)} - \mathbf{y}^{(\mathrm{k})} = -\beta^{(\mathrm{k})}\mathbf{S}^{(\mathrm{k})} \tag{6.77}$$

$$\mathbf{g}^{(\mathrm{k})} = \nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k}+1)} - \nabla\boldsymbol{\mathcal{J}}^{(\mathrm{k})} \tag{6.78}$$

13. Set the new iteration number as $\mathrm{k} = \mathrm{k} + 1$ and go to step 9 [162].

The BFGS algorithm employed for the shape optimization using the Bezier curve can be expressed in a similar way. Instead of $\mathbf{y}$ in Equation (6.75), here the unknowns are $y-$ component of control points, namely B_y.

6.9 Results

In this section, the results obtained for the shape optimization of an airfoil in the incompressible, irrotational, and inviscid flow under given boundary conditions are presented. Three kinds of the design variable (the airfoil grid points, the Bezier curve control points, and the maximum thickness of NACA 00xx airfoils) as well as two optimization methods (CG and BFGS) are considered. In all test cases in this thesis which employ the Bezier curve, the number of predetermined airfoil data, m, is set equal to the Bezier curve order, $n+1$.

Test case 1:

In this test case, the airfoil surface is parameterized by a Bezier curve of order 11. The total number of the design variable is 18, namely, 9 design variables for each of the upper and lower surfaces. At first, two parametric curves for two surfaces (upper and lower) are obtained using 11 grid points and then a fine grid is generated to obtain accurate results. The data used for Test case 1 is given in Table 6-1. The comparison of the initial and optimal airfoil shapes and some magnified parts of them are shown in Figure 6-41. In this test case, BFGS optimization method is employed.

	airfoil	grid size	Angle of attack α	Free stream velocity V_∞
Initial	TsAGI"B" 10%	300×305	1°	$70\frac{\text{m}}{\text{s}}$
Desired	NACA 0015	350×365	1°	$70\frac{\text{m}}{\text{s}}$

Table 6-1 Data used for Test case 1.

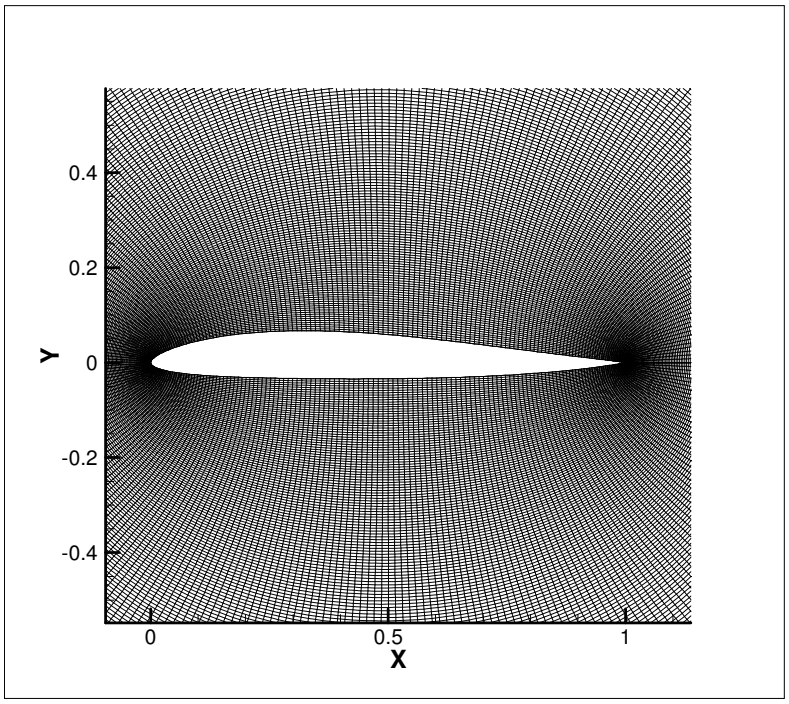


Figure 6-39 Grid used in Test case 1 (around initial shape).

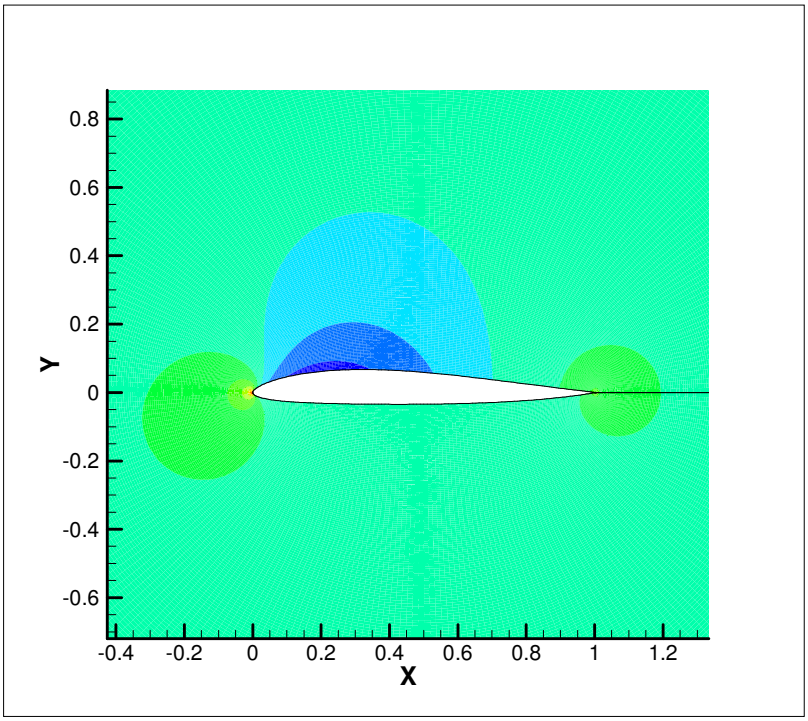


Figure 6-40 Pressure distribution around the airfoil surface (initial shape).

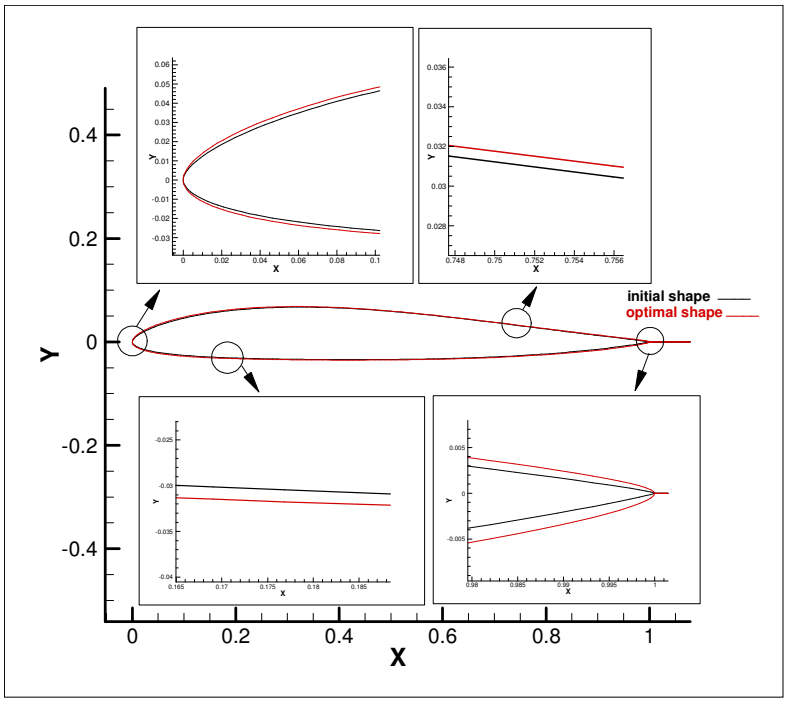


Figure 6-41 Comparison of the initial and optimal shapes and some magnified parts including the leading edge, middle parts, and the trailing edge.

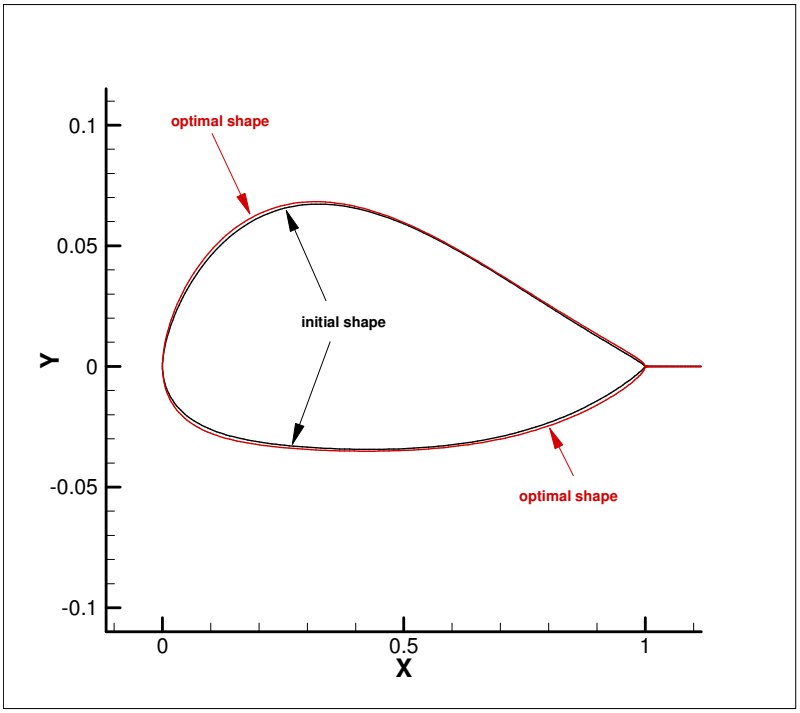


Figure 6-42 Comparison of the initial and optimal shapes. The $y-$axis has been greatly exaggerated to highlight difference in the airfoil shapes.

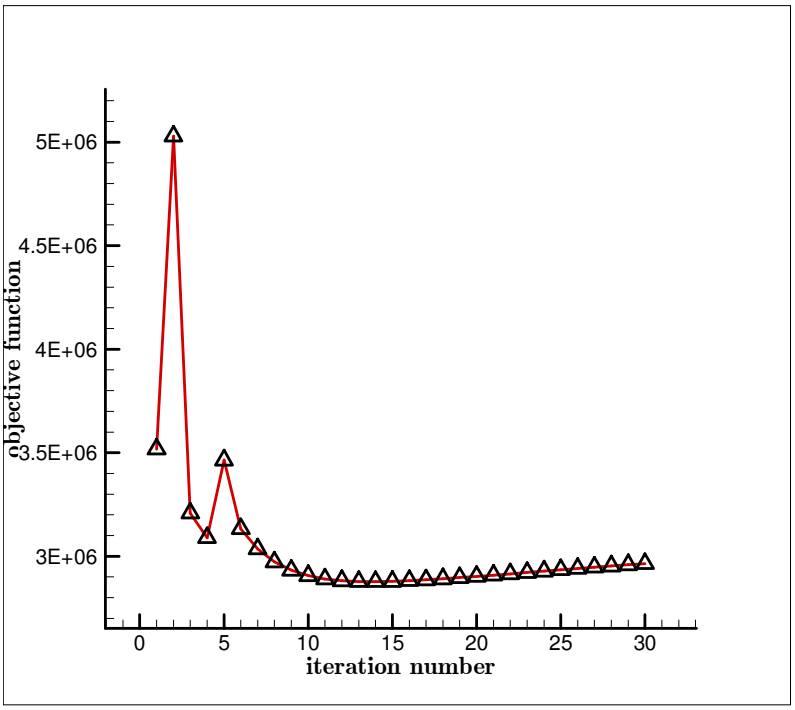


Figure 6-43 Objective function value versus the iteration number.

The convergence of the objective function is shown in Figure 6-43. The initial and minimum values for the objective function are approximately 3517433 and 2877116, respectively, which shows %18.2 reduction in objective function. The minimum value for the objective function takes place in iteration 14. The optimization time spent on the 1st iteration (which is equivalent to one direct flow solution) is 11 minutes & 43 seconds and the total optimization time for 30 iterations is 11 minutes & 46 seconds which shows the proposed sensitivity analysis efficiency. 30 iterations take only 3 seconds. The reason for the difference between the 1st iteration and the following ones is that the solution after the 1st iteration is a very good initial guess for the 2nd iteration and the direct solution converges quickly. In other words, what is a bit time consuming for the 1st iteration is the grid generation and stream function loops not the pressure calculation, sensitivity analysis, and optimization stages. Moreover, a fine grid (300×305) and a tolerance of 10^{-8} are used in the iterative loops which increase the computation time. The code is programmed in Fortran 77 using a Fortran compiler (Force 2.0) and the computations are run on a PC with Intel Pentium Dual 1.73 and 1G RAM. All the computations in the test cases in this chapter are performed

using the above mentioned compiler and PC. Therefore, there is no need to repeat it in the following test cases.

Test case 2:

Test case 2 is similar to Test case 1 but with different specifications. The data for this test case is given in Table 6-2.

	airfoil	grid size	Angle of attack α	Free stream velocity V_∞
Initial	NACA 0028	400×425	2°	$70 \frac{\text{m}}{\text{s}}$
Desired	NACA 0016	350×365	2°	$70 \frac{\text{m}}{\text{s}}$

Table 6-2 Data used for Test case 2.

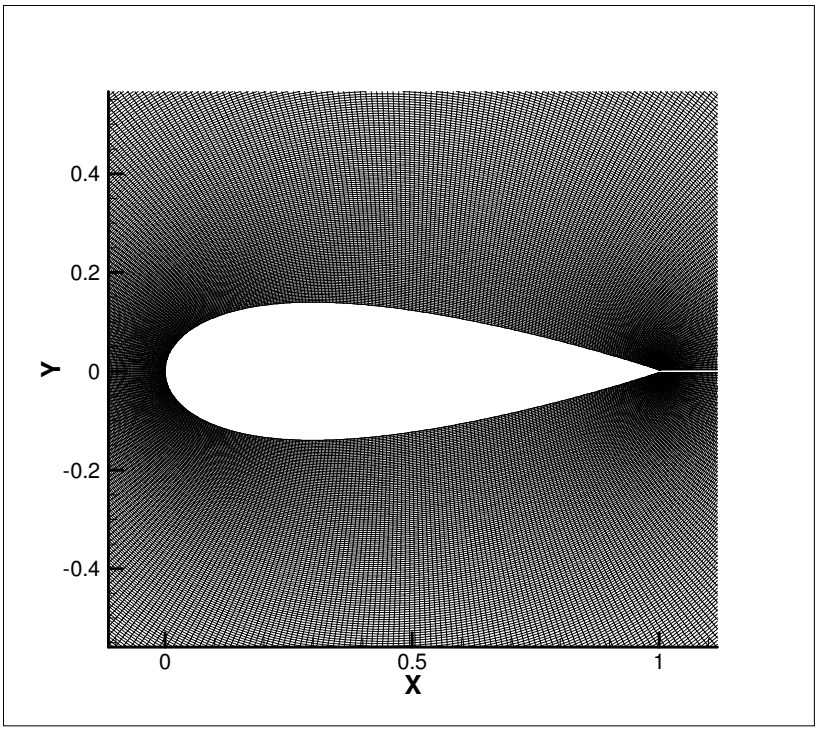


Figure 6-44 Grid used in Test case 2 (around initial shape).

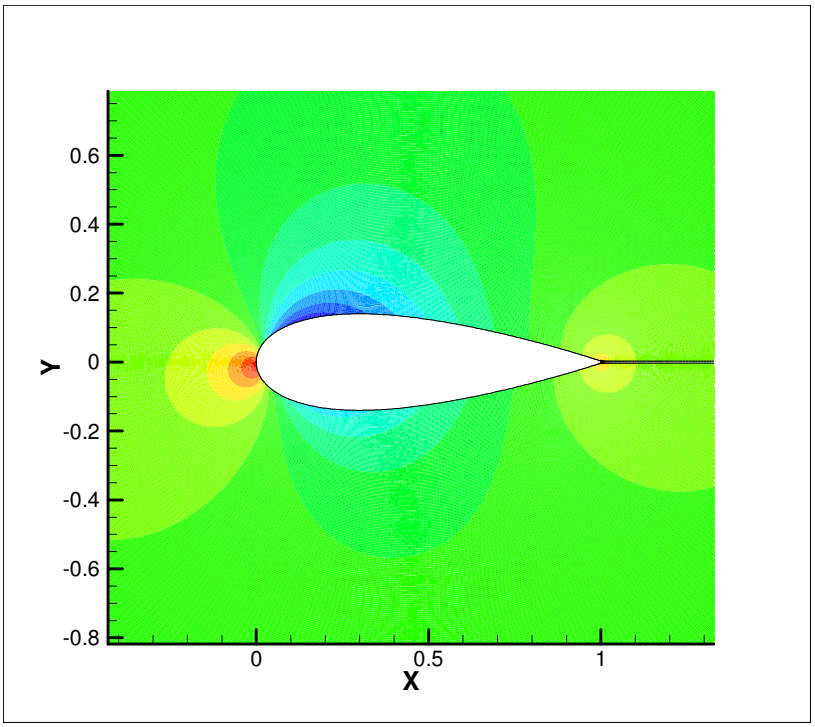


Figure 6-45 Pressure distribution around the airfoil surface (initial shape).

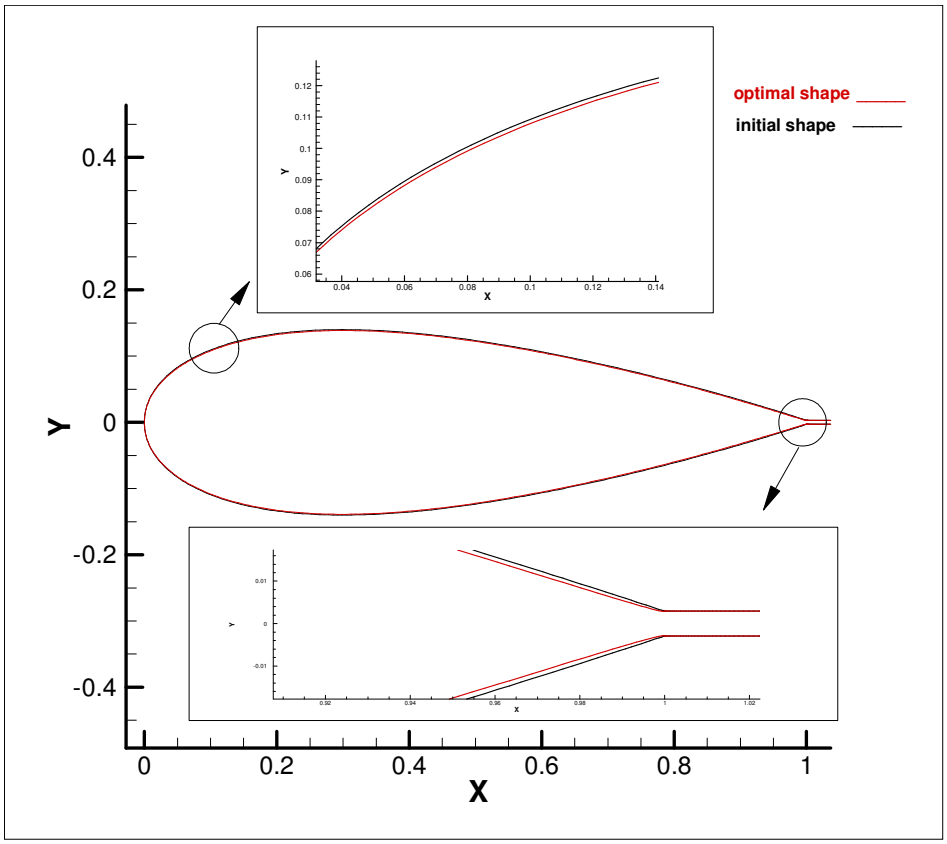


Figure 6-46 Comparison of the initial and optimal shapes and some magnified parts including the leading and trailing edges.

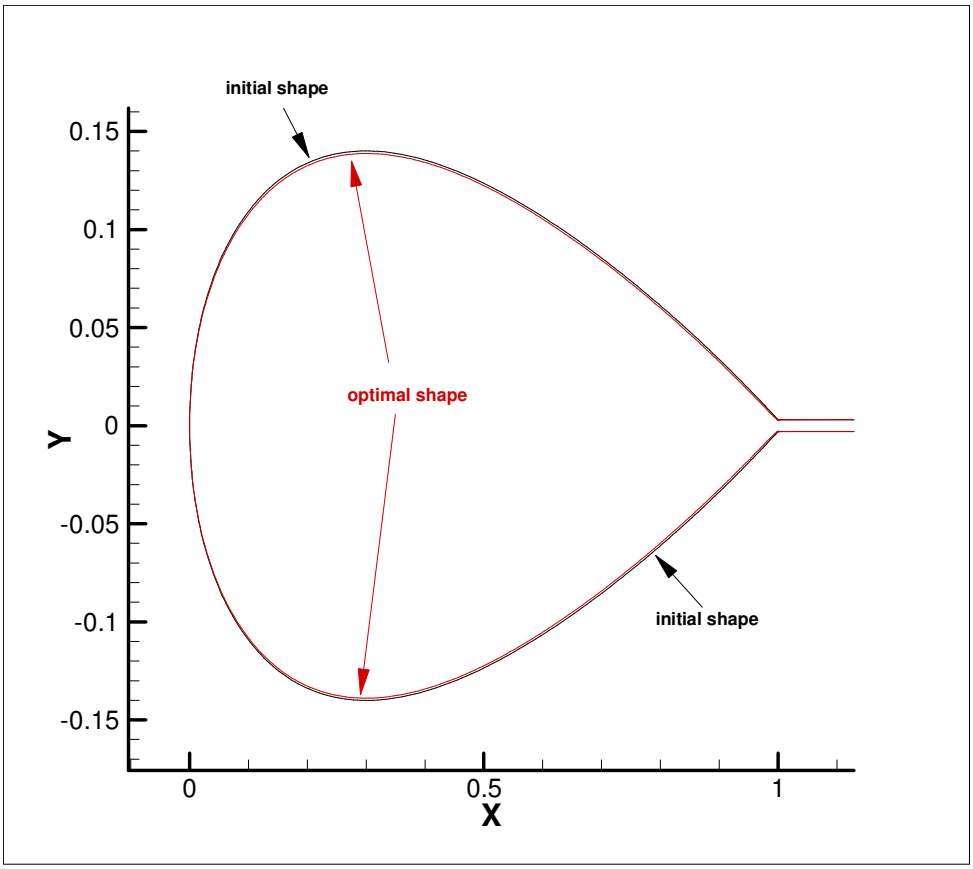


Figure 6-47 Comparison of the initial and optimal shapes. The $y-$axis has been greatly exaggerated to highlight difference in the airfoil shapes.

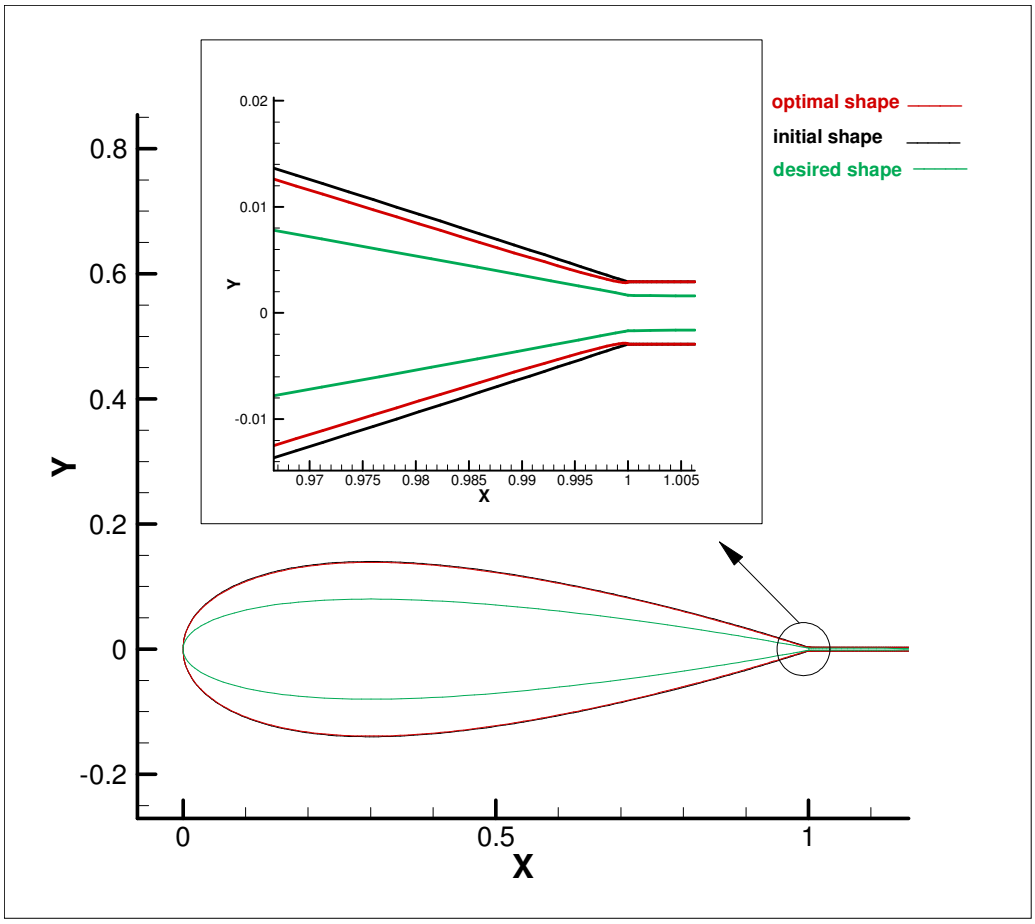


Figure 6-48 Comparison of the initial, optimal, and desired shapes.

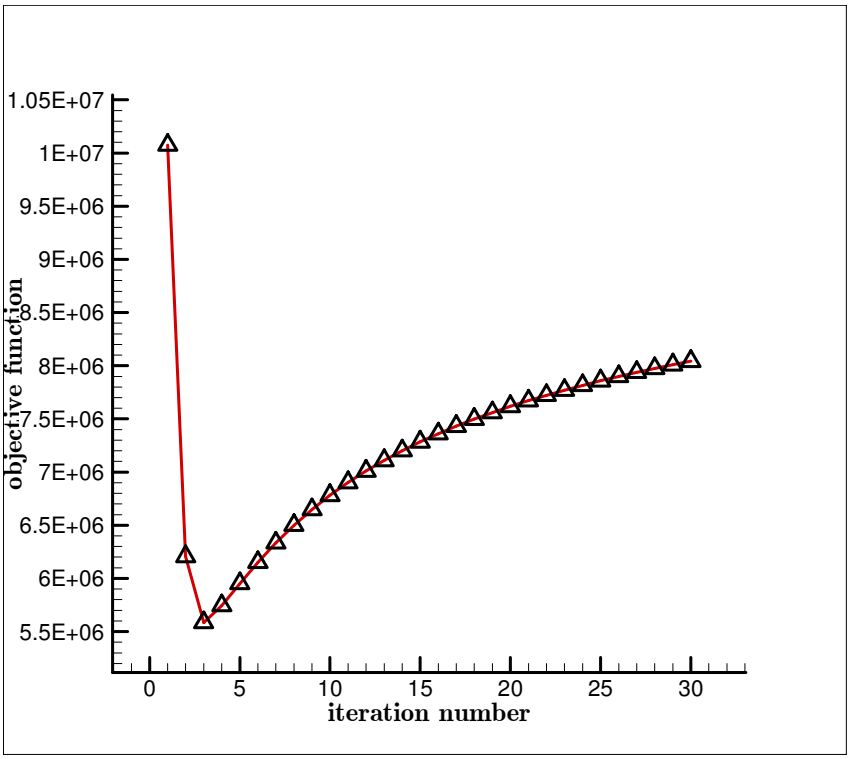


Figure 6-49 Objective function value versus the iteration number.

The explanation is similar to Test case 1. Thus only the results will be given here.

Initial objective function	Minimum objective function (iteration 3)	Computational time, 1st iteration	Total computational time, 30 iterations	Percentage of reduction in objective function
10074507	5585758	35m&30s	35m&34s	%44.5

Table 6-3 Results for Test case 2.

Now an optimal shape design problem using a Bezier curve of order 7 is given. As will be shown, it decreases the objective function value much bigger than when using a Bezier curve of order 11 as there is less number of design variables (10 design variables for a Bezier curve of order 7 versus 18 design variables for a Bezier curve of order 11). However, as will be shown, using a Bezier curve of order 7 is not comprehensive for all airfoil shapes and is suitable to NACA 00XX or similar airfoils only. In other words, it is not able to produce all airfoil shapes precisely.

Test case 3 (using a Bezier curve of order 7):

	airfoil	grid size	Angle of attack α	Free stream velocity V_∞
Initial	NACA 0012	300×305	0°	$70 \frac{\text{m}}{\text{s}}$
Desired	NACA 0017	250×305	0°	$70 \frac{\text{m}}{\text{s}}$

Table 6-4 Data used for Test case 3.

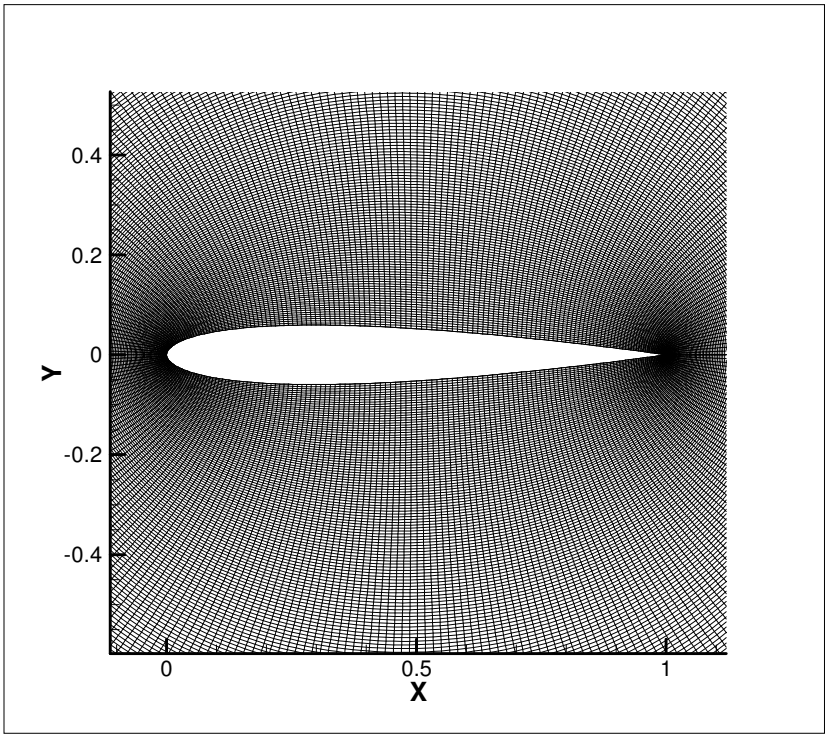


Figure 6-50 Grid used in Test case 3 (around initial shape).

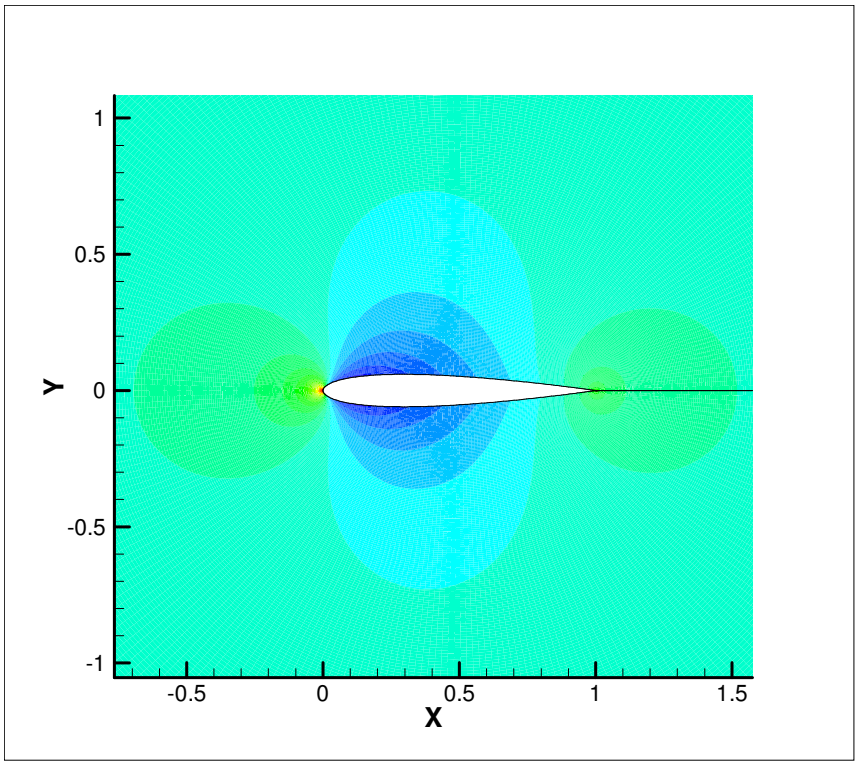


Figure 6-51 Pressure distribution around the airfoil surface (initial shape).

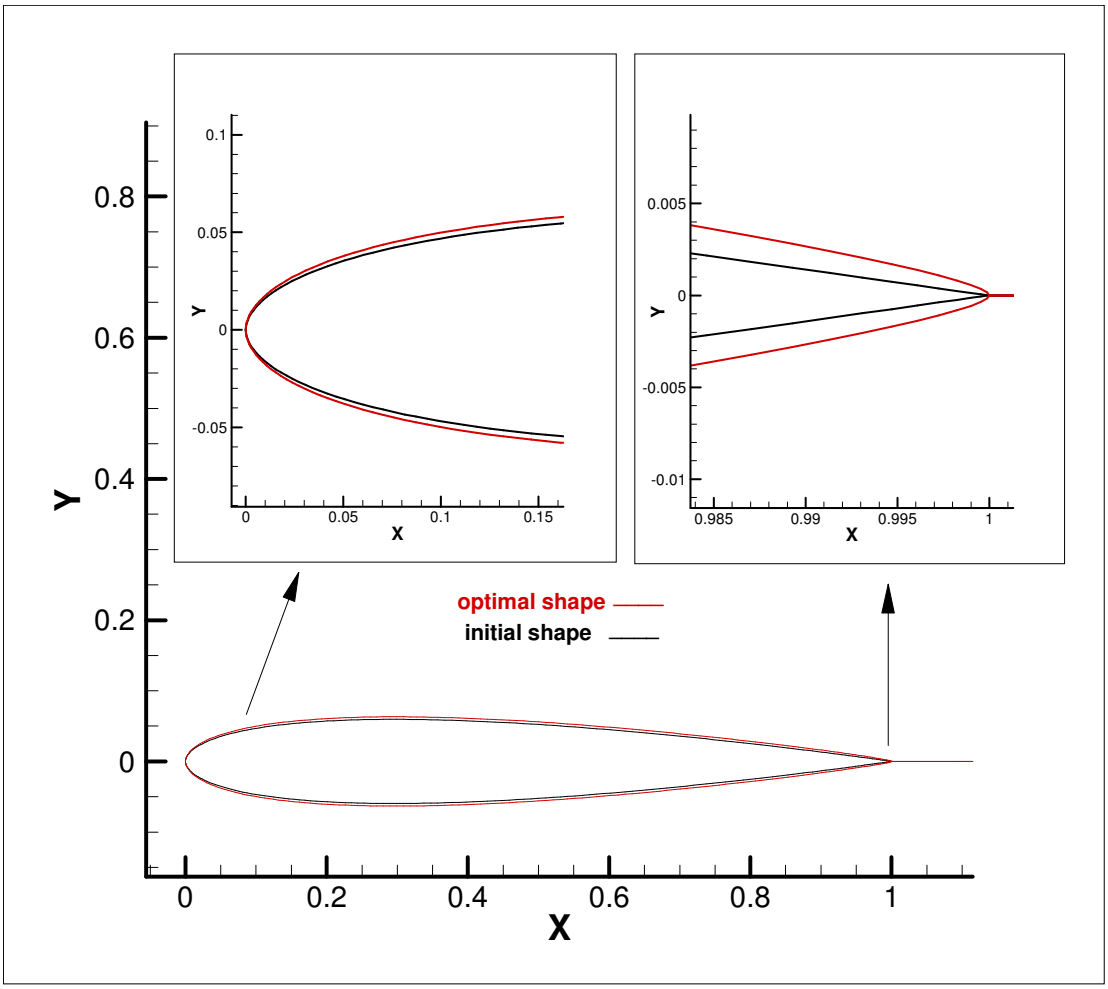


Figure 6-52 Comparison of the initial and optimal shapes and some magnified parts including the leading and trailing edges.

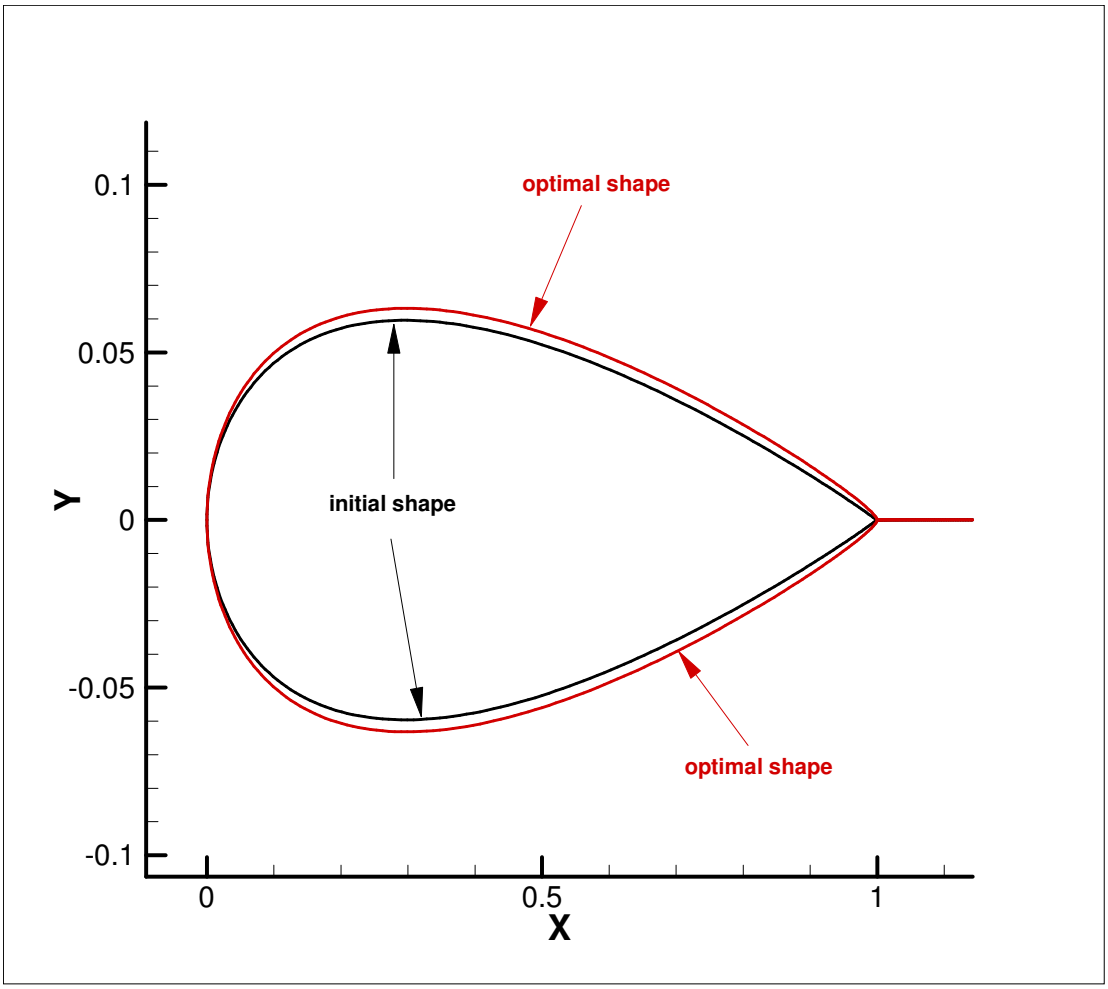


Figure 6-53 Comparison of the initial and optimal shapes. The $y-$ axis has been greatly exaggerated to highlight difference in the airfoil shapes.

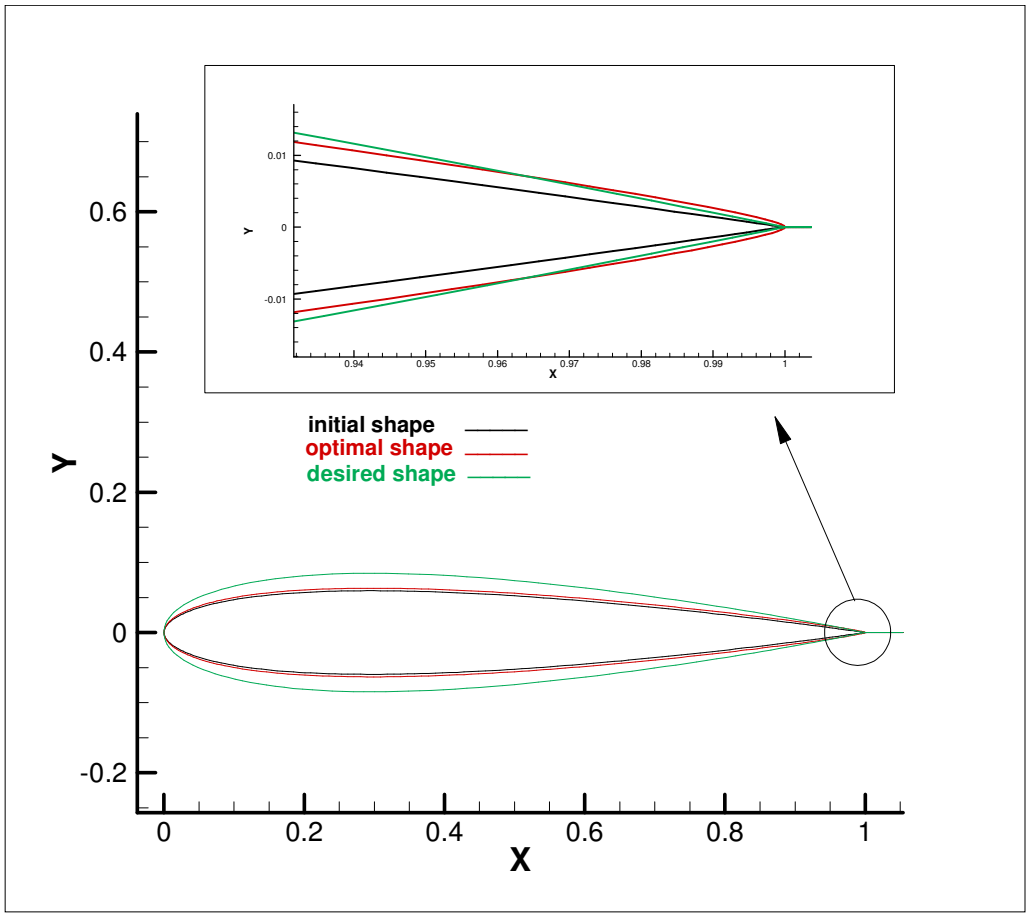


Figure 6-54 Comparison of the initial, optimal, and desired shapes.

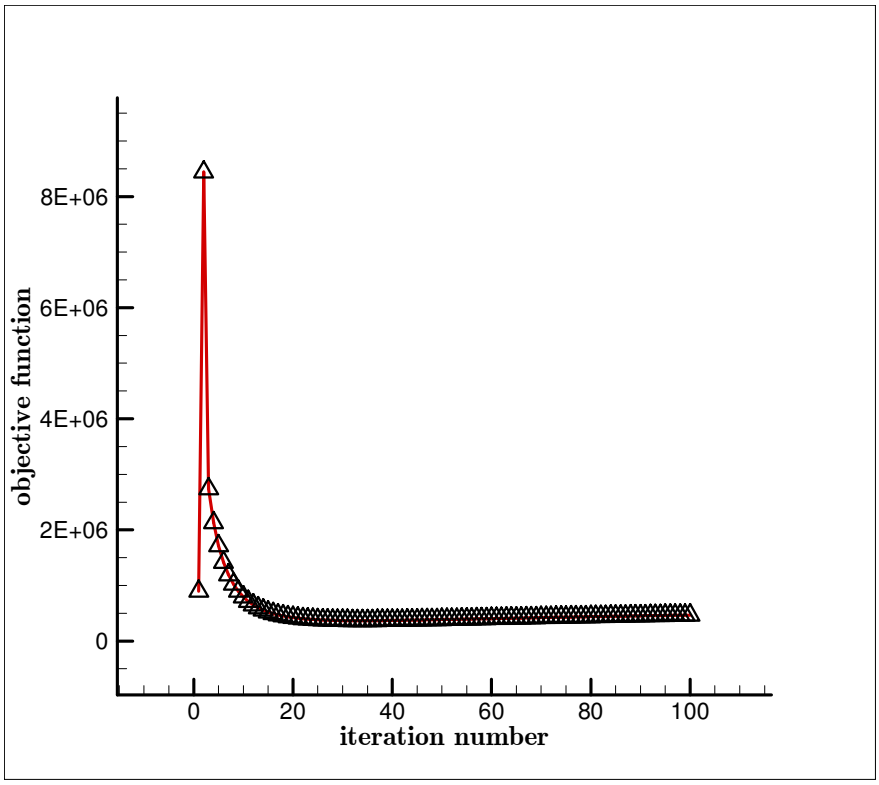


Figure 6-55 Objective function versus iteration number.

The results are given in Table 6-5.

Initial objective function	Minimum objective function (iteration 34)	Computational time, 1st iteration	Total computational time, 100 iterations	Percentage of reduction in objective function
898639	368148	9m&32s	9m&43s	%59

Table 6-5 Results for Test case 3

Although there is a decrease of %59 in objective function, the %59 approach from the initial shape to desired one is not seen (see Figure 6-54). This indicates that the solution of the inverse problem is not unique. The reason for this can be found in the trailing edge configuration for the initial, optimal, and desired airfoil shapes (Figure 6-54).

Although the results of using the Bezier curve of order 7 is very promising, its drawback is that it is restricted to the simple and symmetric airfoil shapes such as NACA 00XX. For other airfoil shapes, there can be seen some oscillations around the trailing edge (see Figure 6-56).

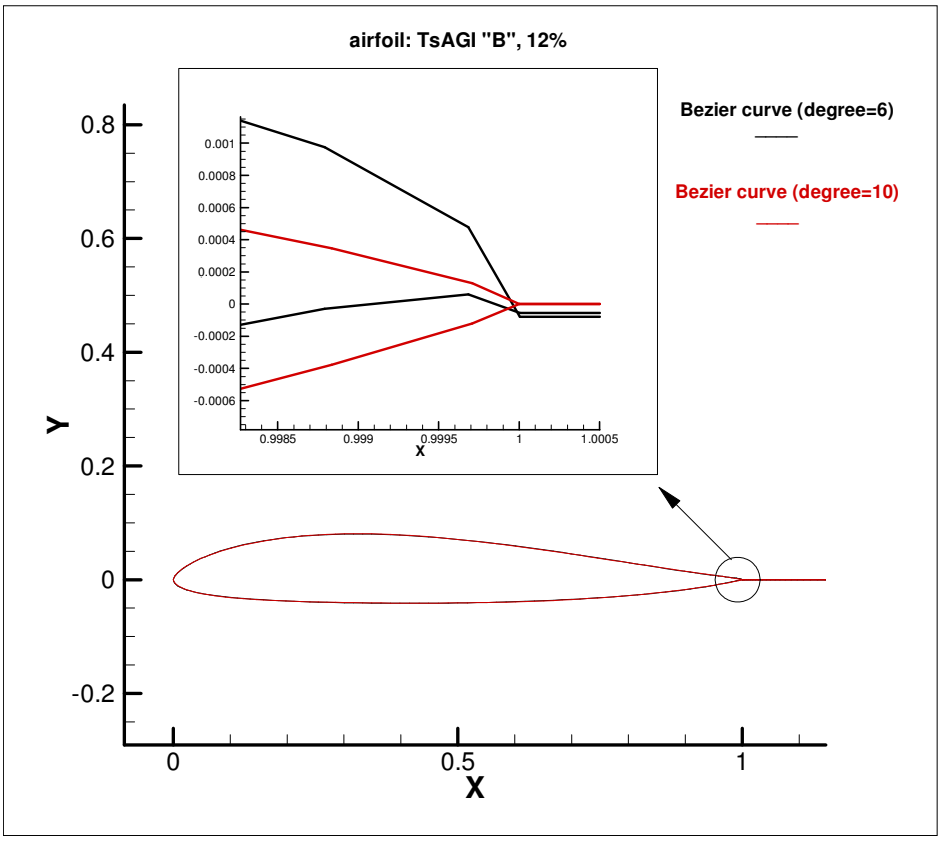


Figure 6-56 Oscillations around the trailing edge.

Moreover, the Bezier curve of order 7 fails to represent the NACA 00XX airfoils accurately. In other words, the Bezier curve of order 11 is the appropriate option to produce very accurate airfoils (Figure 6-57).

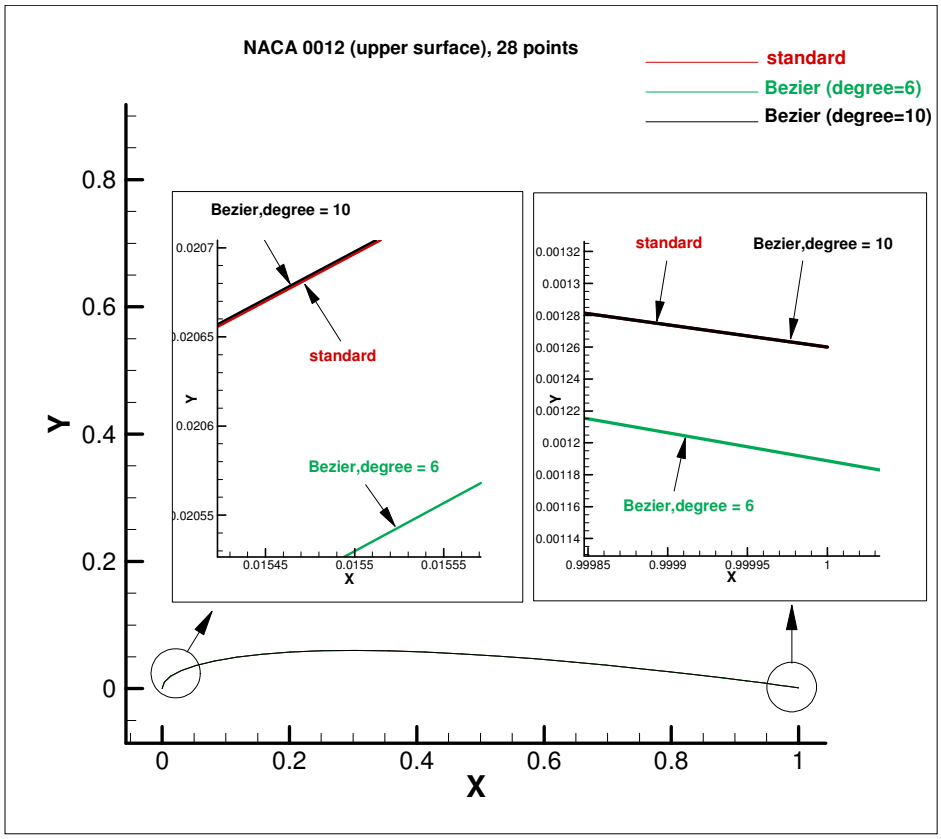


Figure 6-57 Comparison of an analytical NACA 0012 airfoil (upper surface only) with one obtained by using the Bezier curves of order 7 and 11. The plots represent an excellent agreement between the analytical NACA and the Bezier of order 11.

Test case 4:

In this test case, the airfoil surface is parameterized by grid points obtained from the analytical NACA formula (e.g. software JavaFoil). In this case, the number of the design variables is equal to the number of grid points minus three (one for leading edge and two for trailing edge). Therefore, we have an aerodynamic shape optimization problem with a high number of the design variables as we should have a fine grid to obtain sufficiently accurate results. It is known that the optimization process becomes more challenging by increasing the number of design variables. Hence we have a difficult shape optimization problem in Test case 4. The data used for Test case 4 is given in Table 6-6. A very fine grid (400×425) is used for the initial airfoil shape (NACA 0012). The number of the design variables is $N-3$ which is $425-3=422$. Therefore, a time consuming optimization problem is

expected. However, by using the sensitivity analysis scheme proposed in this chapter, the total time for the optimization problem in Test case 4 using both CG and BFGS optimization methods is about 46 minutes for 8000 iterations. The computation time for the 1st iteration is about 25 minutes. The comparison of the computation times for the 1st iteration and 8000 ones indicates again the efficiency of the sensitivity analysis scheme. The summary of the results is presented in Table 6-7. The comparison of the initial, optimal, and desired airfoil shapes is given in Figure 6-58. As can be seen in the figure, the variation of the shape is minute. The convergence of the objective function to a local minimum when both the CG and BFGS optimization methods are used as well as a comparison of them are shown in Figure 6-59 to Figure 6-61, respectively. The plots reveal the better performance of the BFGS method in minimizing the objective function.

	airfoil	grid size	Angle of attack α	Free stream velocity V_∞
Initial	NACA 0012	400×425	0°	$70 \frac{\text{m}}{\text{s}}$
Desired	NACA 0014	400×425	0°	$70 \frac{\text{m}}{\text{s}}$

Table 6-6 Data used for Test case 4

	Initial $\mathcal{J}$	Final $\mathcal{J}$	Number of iterations	Computation time(total)	Reduction in $\mathcal{J}$
CG	6351997	6053996	8000	~46mins	%4.7
BFGS	6351997	6036943	8000	~46mins	%5

Table 6-7 Summary of results for Test case 4

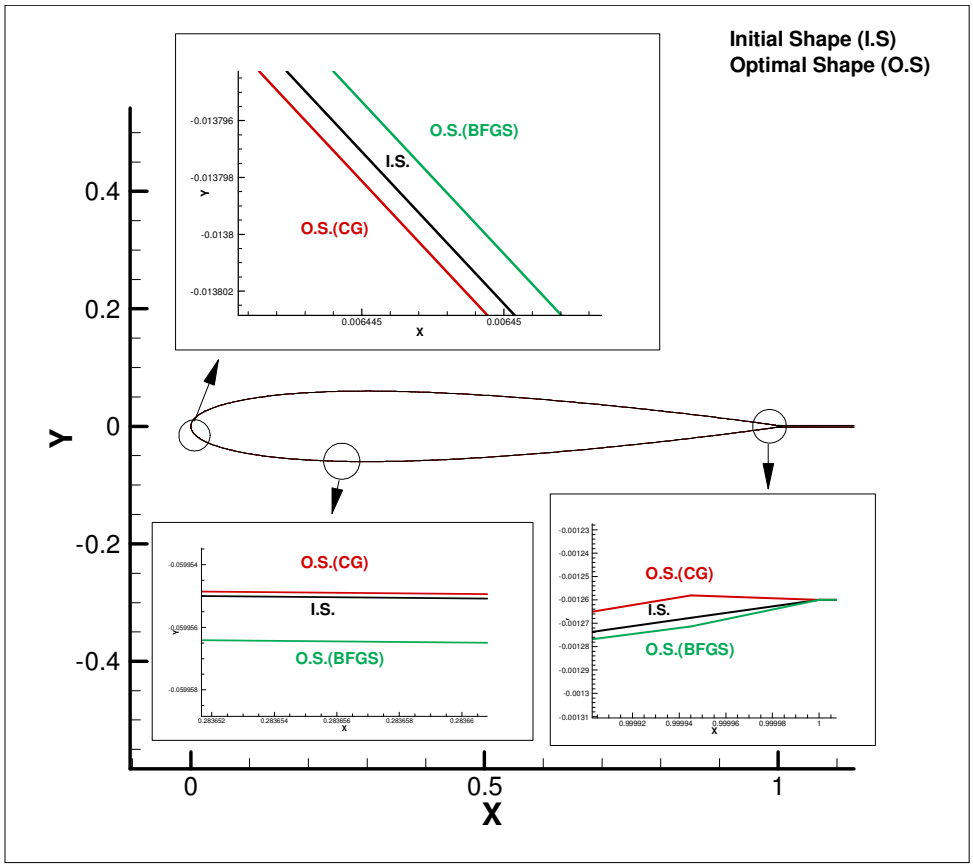


Figure 6-58 Comparison of the initial and optimal airfoil shapes with the magnified sections of them to show the variation of the shape. Both BFGS and CG are used in optimization process.

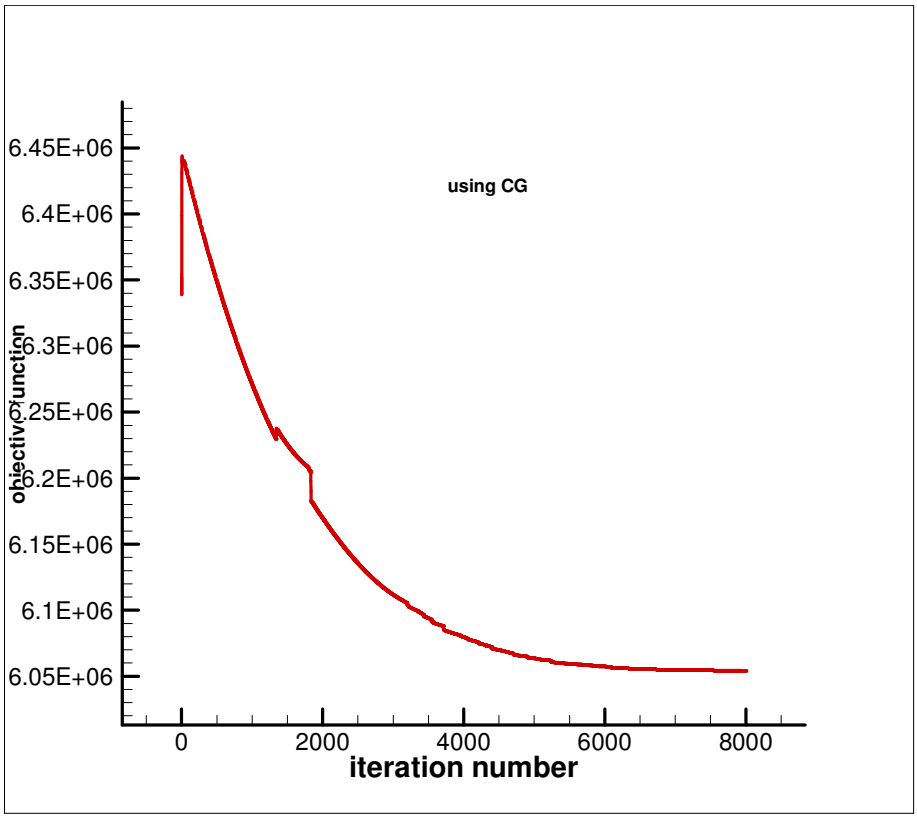


Figure 6-59 Convergence of the objective function. CG is used as the optimization method.

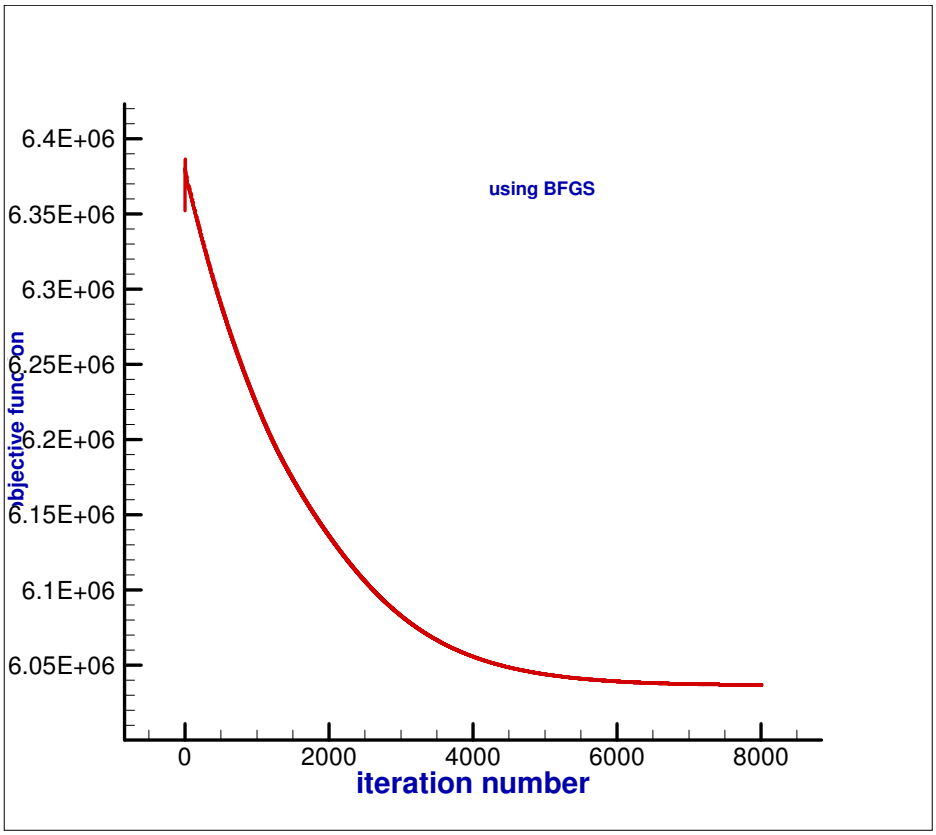


Figure 6-60 Convergence of the objective function. BFGS is used as the optimization method.

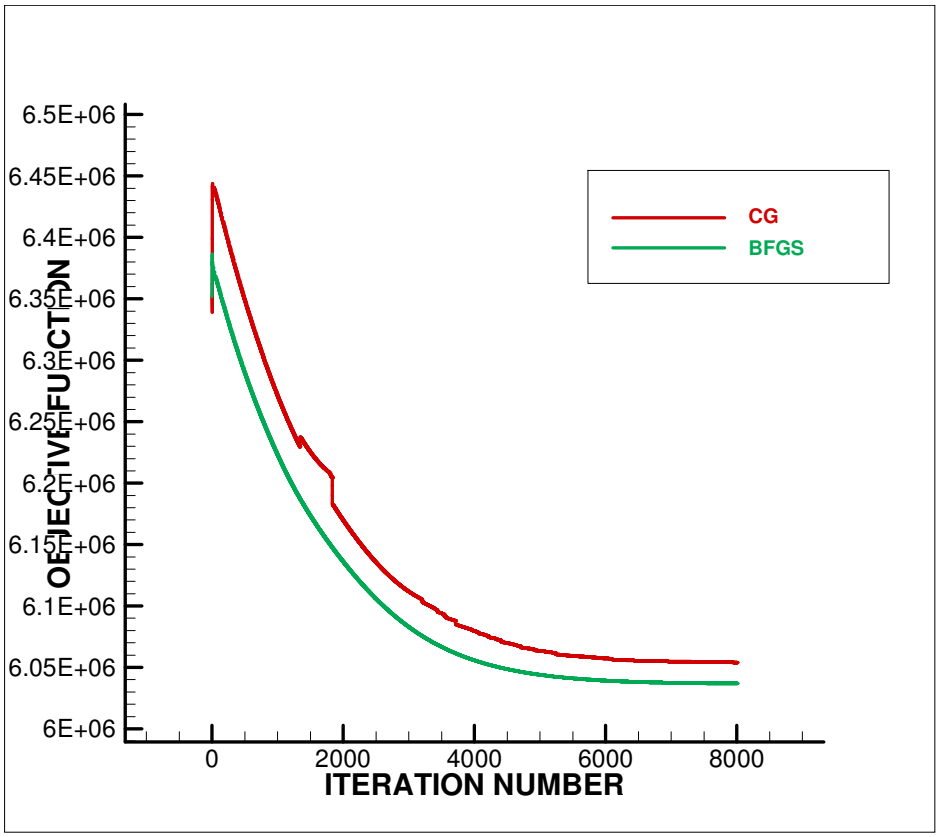


Figure 6-61 Comparison of the CG and BFGS methods in decreasing the objective function.

In Test cases 5 and 6 the maximum thickness of the NACA 00xx is considered as a design variable. As mentioned previously, the conjugate gradient method is employed as the optimization method.

Test case 5:

The data used for the problem including the conditions for the initial and desired airfoils are given in Table 6-8.

	airfoil	grid size	Angle of attack α	Free stream velocity V_∞
Initial	NACA 0011	80×125	2°	$70 \frac{\text{m}}{\text{s}}$
Desired	NACA 0018	80×125	2°	$70 \frac{\text{m}}{\text{s}}$

Table 6-8 Data used for Test case 5

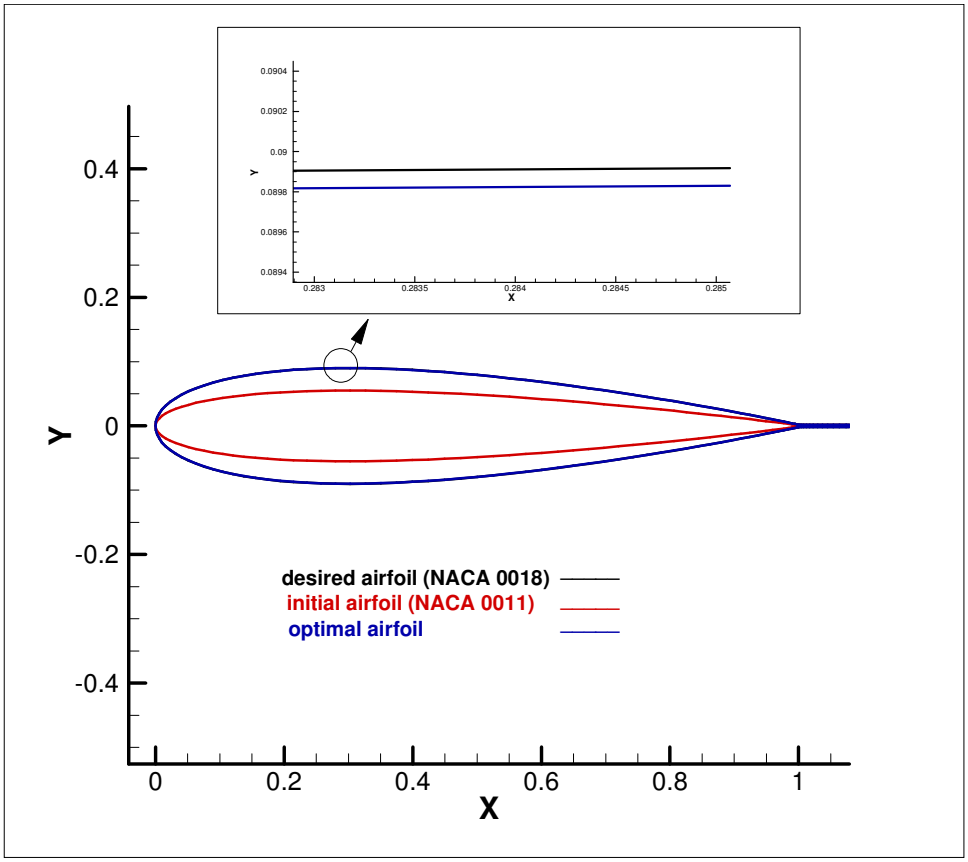


Figure 6-62 The initial, optimal, and desired shapes for the airfoil. There is an excellent agreement between the optimal and desired airfoil shapes.

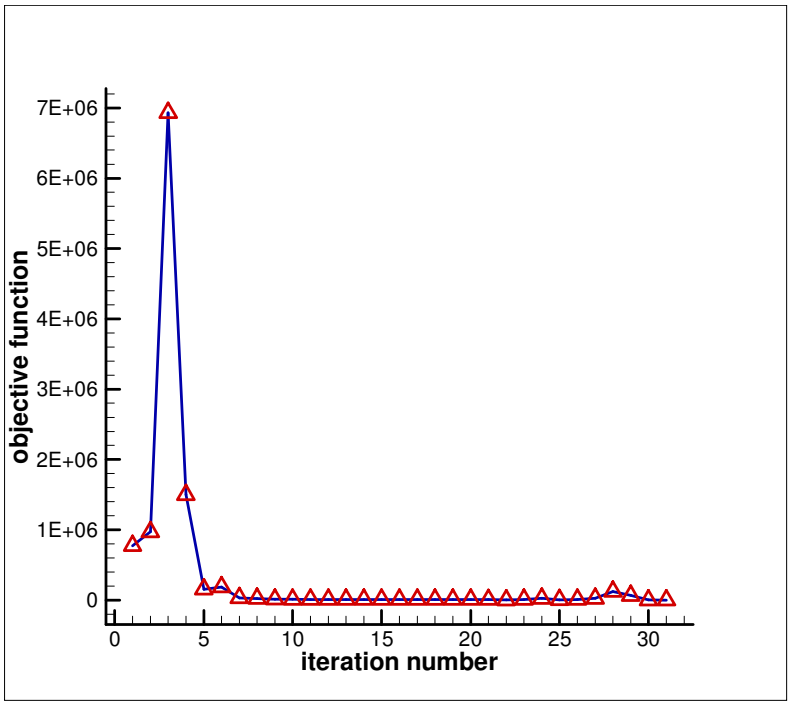


Figure 6-63 Objective function versus iteration number.

	Initial $\mathcal{J}$	Final $\mathcal{J}$	Number of iterations	Computation time(total)	Reduction in $\mathcal{J}$
CG	773597.45	15.48	31	2m:14s	~ %100

Table 6-9 Results for Test case 5

Figure 6-62 represents the comparison of the initial, optimal, and desired shapes for airfoils. The desired airfoil shape is a NACA0018 at conditions stated in Table 6-8. As can be seen, this shape is shown by a black color line. The optimization process is started by a NACA0011 as an initial shape which is shown by a red color line. The optimal shape (shown by a blue color line), which is obtained by the conjugate gradient method, is in an excellent agreement (full matching) with the desired one. The objective function variation is shown in Figure 6-63. The initial and final values for the objective function are about 773597.45 and 15.48, respectively, which reveals an approximately %100 reduction in the objective function within 31 iterations. The total time for the optimization (for 31 iterations) is 2 minutes and 14

seconds. The tolerance used in iterative steps in the program is 10^{-8}. Although such a tolerance value increases the computation time, it enhances the accuracy of the results. If the $y-$ components of the maximum thickness in upper and lower airfoil surfaces are denoted by y_{thickU} and y_{thickL}, respectively, the value of the pressure for these two locations for initial, optimal, and desired shapes are reported in Table 6-10. The difference values show the validity of the shape optimization process.

	Initial shape	Optimal shape	Desired shape	Difference (Pa)
Pressure at y_{thickU} (Pa)	100080.04	99411.37	99410.06	Diff. in In. and Des. = 669.98 Diff. in Opt. and Des. = 1.31
Pressure at y_{thickL} (Pa)	100814.39	100248.25	100244.54	Diff. in In. and Des. = 569.85 Diff. in Opt. and Des. = 3.71

Table 6-10 Comparison of the pressure at the maximum thicknesses of the airfoil surface (upper and lower surfaces) for the initial, optimal, and desired shapes.

Test case 6:

The data for the problem is given in Table 6-11. The explanation for the results is similar to Test case 5.

	airfoil	grid size	Angle of attack α	Free stream velocity V_∞
Initial	NACA 0015	80×125	2°	$70\frac{\text{m}}{\text{s}}$
Desired	NACA 0035	80×125	2°	$70\frac{\text{m}}{\text{s}}$

Table 6-11 Data used for Test case 6

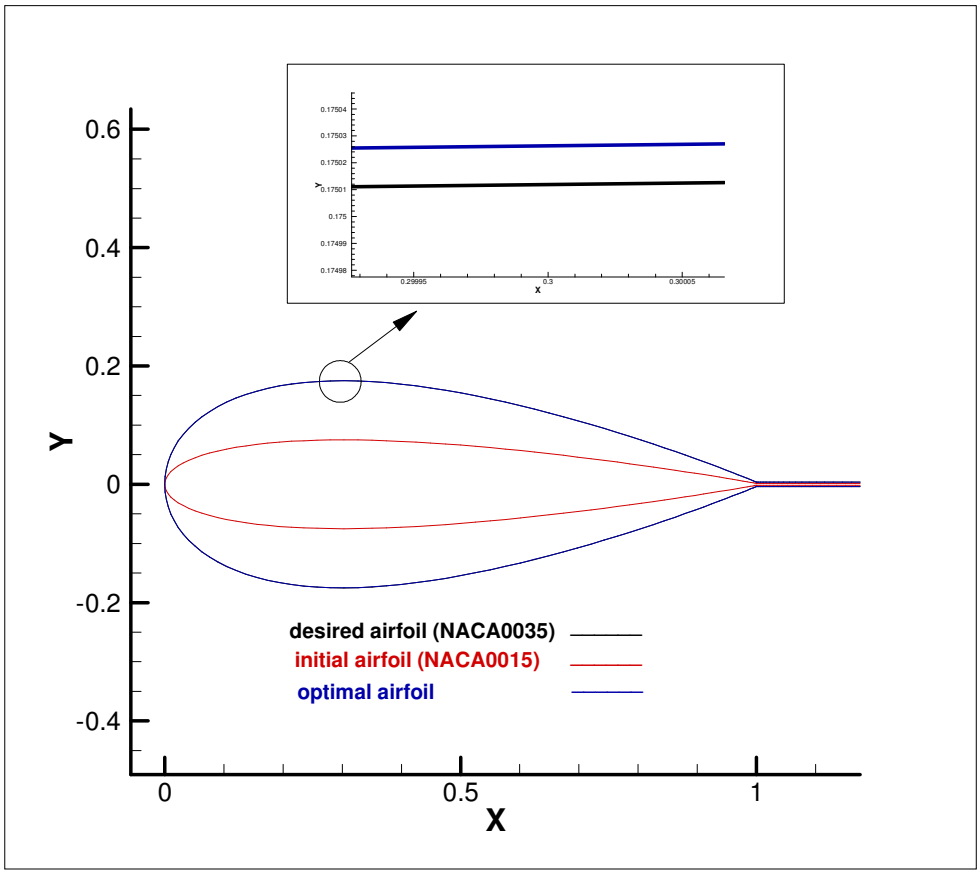


Figure 6-64 The initial, optimal, and desired shapes for the airfoil. There is an excellent agreement between the optimal and desired airfoil shapes.

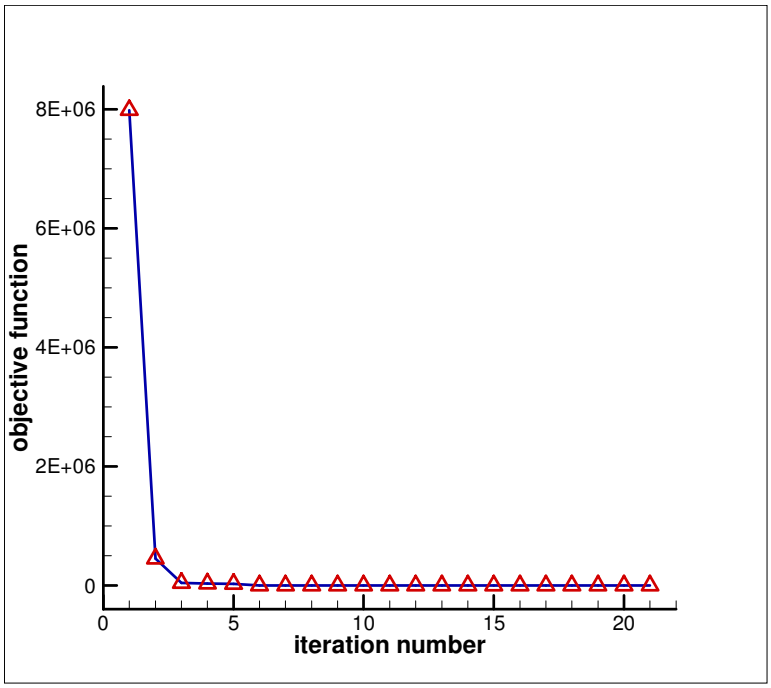


Figure 6-65 Objective function value versus the iteration number.

	Initial $\mathcal{J}$	Final $\mathcal{J}$	Number of iterations	Computation time(total)	Reduction in $\mathcal{J}$
CG	7986880.34	1.15	21	1m:08s	~ %100

Table 6-12 Results for Test case 6

	Initial shape	Optimal shape	Desired shape	Difference (Pa)
Pressure at y_{thickU} (Pa)	99703.83	97552.39	97552.99	Diff. in In. and Des. = 2150.84 Diff. in Opt. and Des. = 0.60
Pressure at y_{thickL} (Pa)	100494.52	98662.17	98661.28	Diff. in In. and Des. = 1833.24 Diff. in Opt. and Des. = 0.89

Table 6-13 Comparison of the pressure at the maximum thicknesses of the airfoil surface (upper and lower surfaces) for the initial, optimal, and desired shapes.

6.10 Adjoint method

As pointed out previously, for the aerodynamic shape optimization problems requiring a large number of design variables, the use of finite difference method to evaluate the gradient by introducing a small perturbation to each design variable separately and then solving the flow problem is of very high computational cost. Because it requires a number of additional flow solution equal to the number of design variables. For optimal shape design problems with a high number of design variables, the adjoint method [106] can compute the gradients of objective function much faster than the finite difference method.

The aerodynamic shape optimization problem of interest here can be expressed as

$$\begin{aligned} &\text{minimization of objective function } \boldsymbol{\mathcal{J}} \\ &\text{subject to constraint } \boldsymbol{\mathcal{R}} = 0 (\text{the governing equation}) \end{aligned} \tag{6.79}$$

The objective function $\boldsymbol{\mathcal{J}}$ and the governing equation $\boldsymbol{\mathcal{R}} = 0$ depend on the flow variables $\mathbf{W}$ and the geometry design variable $\mathbf{X}_D$:

$$\boldsymbol{\mathcal{J}} = \boldsymbol{\mathcal{J}}(\mathbf{W}, \mathbf{X}_D) \tag{6.80}$$

$$\boldsymbol{\mathcal{R}} = \boldsymbol{\mathcal{R}}(\mathbf{W}, \mathbf{X}_D) = 0 \tag{6.81}$$

The derivative of the objective function $\boldsymbol{\mathcal{J}}$ with respect to the design variables $\mathbf{X}_D$ can be expressed as

$$\frac{d\boldsymbol{\mathcal{J}}}{d\mathbf{X}_D} = \frac{\partial \boldsymbol{\mathcal{J}}}{\partial \mathbf{X}_D} + \frac{\partial \boldsymbol{\mathcal{J}}}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial \mathbf{X}_D} \tag{6.82}$$

which states that a change in the objective function is due to a combination of a variation in the flow solution $\partial \mathbf{W}$ and a variation in the design variable (change in geometry) $\partial \mathbf{X}_D$. In a similar way, we have

$$\frac{d\boldsymbol{\mathcal{R}}}{d\mathbf{X}_D} = \frac{\partial \boldsymbol{\mathcal{R}}}{\partial \mathbf{X}_D} + \frac{\partial \boldsymbol{\mathcal{R}}}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial \mathbf{X}_D} = 0 \tag{6.83}$$

If the sensitivity analysis is carried out using Equations (6.82) and (6.83), the problem is referred to as the "primal problem". Solving the primal problem comes with the same difficulties as we encounter with use of the finite difference method. It requires the additional flow solutions proportional to the number of the design variables $\mathbf{X}_D$. Therefore, the adjoint method comes to the picture by introducing a vector of Lagrange multipliers Ψ. By adding Equation (6.83) as a constraint to the sensitivity equation (Equation (6.82), we obtain

$$\frac{d\boldsymbol{\mathcal{J}}}{d\mathbf{X}_D} = \frac{\partial \boldsymbol{\mathcal{J}}}{\partial \mathbf{X}_D} + \frac{\partial \boldsymbol{\mathcal{J}}}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial \mathbf{X}_D} - \Psi^{\mathrm{T}} \left\{ \overbrace{\frac{\partial \boldsymbol{\mathcal{R}}}{\partial \mathbf{X}_D} + \frac{\partial \boldsymbol{\mathcal{R}}}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial \mathbf{X}_D}}^{=0} \right\} \tag{6.84}$$

Rearranging the terms inside Equation (6.84), we get

$$\frac{d\boldsymbol{\mathcal{J}}}{d\mathbf{X}_D} = (\frac{\partial \boldsymbol{\mathcal{J}}}{\partial \mathbf{X}_D} - \Psi^{\mathrm{T}} \frac{\partial \boldsymbol{\mathcal{R}}}{\partial \mathbf{X}_D}) + (\frac{\partial \boldsymbol{\mathcal{J}}}{\partial \mathbf{W}} - \Psi^{\mathrm{T}} \frac{\partial \boldsymbol{\mathcal{R}}}{\partial \mathbf{W}}) \frac{\partial \mathbf{W}}{\partial \mathbf{X}_D} \tag{6.85}$$

If

$$\frac{\partial \boldsymbol{\mathcal{J}}}{\partial \mathbf{W}} - \Psi^{\mathrm{T}} \frac{\partial \boldsymbol{\mathcal{R}}}{\partial \mathbf{W}} = 0 \tag{6.86}$$

Then Equation (6.85) reduces to

$$\frac{d\boldsymbol{\mathcal{J}}}{d\mathbf{X}_D} = \frac{\partial \boldsymbol{\mathcal{J}}}{\partial \mathbf{X}_D} - \Psi^{\mathrm{T}} \frac{\partial \boldsymbol{\mathcal{R}}}{\partial \mathbf{X}_D} \tag{6.87}$$

Equation (6.86) is the *adjoint equation* and the vector Ψ is the *adjoint variables*. Equations (6.86) and (6.87) are referred to as the "dual problem". The adjoint equation is a linear system and can be solved to obtain Ψ. Then the determined Ψ can be substituted into Equation (6.87) to obtain the gradient of the objective function. It can be seen that the gradient of the objective function can be determined without the need for additional flow solutions. The computational cost of solving the adjoint equation is comparable to that of solving the flow equation. Therefore, the computational cost of evaluating the objective function gradient is roughly equal to the

computational cost of two flow equation solutions, independent of the number of design variables [163-166].

From the accuracy of the derivatives view point, the finite difference method (based on the perturbation scheme) is compared to the adjoint method [167-169]. The comparison shows a very good agreement between two methods. Therefore, our aim here is to compare our novel shape sensitivity method to the adjoint method from the efficiency viewpoint only. As mentioned above, the computational cost of solving the adjoint equation is comparable to that of solving the flow equation whereas the computational cost of our novel method is comparable to that of computation of an algebraic expression for arrays of a matrix. As seen in Test case 4, the computation time for iterations 2 to 8000 is $46-25=21$ minutes. In other words, about 7 iterations per second that reveals the efficiency of the proposed sensitivity analysis.

6.11 Conclusion

This chapter addressed the aerodynamic shape optimization for an airfoil in an irrotational and incompressible flow governed by Laplace equation using a type of the elliptic grid generation (O-type), a novel and very efficient sensitivity analysis method, and the conjugate gradient and BFGS optimization methods. The airfoil was parameterized using the grid points and the Bezier curve. Three different types of design variable were considered: The grid points, the Bezier curve control points, and the maximum thickness of NACA00xx airfoils. It was represented that the use of the Bezier curve significantly improves the optimization performance to reach the optimal shape. The results obtained in test cases presented in this chapter show that the proposed sensitivity analysis method reduces the computation cost even for large number of the design variables (Test case 4) and confirm accuracy and efficiency of the proposed shape optimization algorithm.

7. Closing Remarks: Conclusions and Recommendations

This final chapter outlines the main conclusions which may be drawn from the proposed numerical algorithm for the optimal shape design and recommends some further research based on the proposed algorithm.

7.1 Conclusions

In this research work, the optimal shape design in both heat transfer (2D and 3D) and aerodynamics (2D) was investigated using the body fitted grid generation (both algebraic and elliptic grid generation techniques) to mesh the body, a novel and very efficient sensitivity analysis based on the finite difference method to calculate the shape sensitivity coefficients, and gradient based optimization methods (Levenberg-Marquardt, Conjugate Gradient, and Quasi-Newton) to minimize the objective function and infer the optimal shape design.

In both heat transfer and aerodynamics parts, an objective function was defined as least squares norm and a novel and very efficient method was proposed to compute the sensitivity coefficients. The main contribution of the thesis lies in this section. Unlike the perturbation method, this novel method computes all sensitivity coefficients in only one solution of the direct problem (at each iteration). Therefore, it is very efficient for problems involving large number of variables. In fact, it takes advantage of both FDM simplicity and adjoint method efficiency (from computation time point of view). As was seen, the sensitivity analysis for 3D problems using the perturbation method

was very time consuming and computationally intensive process. Our proposed method performed this job in a very efficient manner.

Then three gradient based optimization methods, namely, Levenberg-Marquardt, conjugate gradient, and Quasi-Newton ones were used to minimize the objective function value and obtain the optimal solution. Two validation test cases were given to validate the proposed shape optimization algorithm with the analytical solutions. In heat transfer part, two optimization methods, namely, Levenberg-Marquardt and conjugate gradient were used. Also, both elliptic and TFI grid generation methods were used to mesh the physical domain. The numerical results confirmed that the optimal shape design using the elliptic grid generation technique and the conjugate gradient method is much preferable (see Test case 5 (CGM) in Chapter 3). Other combinations, namely, elliptic grid generation method and Levenberg-Marquardt and TFI grid generation method and the conjugate gradient one also represented very good performance. In aerodynamics part, the conjugate gradient and Quasi-Newton methods were used and the numerical results revealed a very good performance for both optimization methods. In the aerodynamics shape optimization problems, a combination of TFI and elliptic grid generation methods were used to mesh the physical domain. In this scheme, the obtained grid using TFI acted as an initial guess for the elliptic grid generation method. This scheme resulted in very smooth grid spacing around the airfoil body.

The objective function value in demonstrated Test cases is reduced quickly by several orders of magnitude in heat transfer part and is decreased significantly in aerodynamics part, indicating that the proposed optimal shape design algorithm is very accurate, robust, and efficient.

7.2 Recommendations for future work

This final chapter addresses some recommendations for the further study in the framework of the research work presented in this thesis.

1. In heat transfer part, only the steady state heat conduction was considered and both 2D and 3D shape optimization in transient heat conduction may be subject to further research. In other words, the research carried out in heat

transfer part can be repeated for which the steady state heat conduction equation is replaced by the transient heat conduction one.

2. In heat transfer part, only convective boundary condition was considered for the shape optimization problems. Therefore, implementation of the radiation or combination of convective/radiation boundary conditions may be of further work.

3. Grid generation step in three-dimensional shape optimization in heat transfer was carried out only by the elliptic grid generation technique. The use of three-dimensional algebraic grid generation methods (TFI) allows for the implementation of finer grids for the physical domain, thereby reducing the computational cost significantly.

4. In heat transfer part, only the Levenberg-Marquardt and the Conjugate Gradient optimizations methods were employed. The use of other optimization methods such as BFGS may be necessary to compare merits of these methods.

5. The objective function in shape optimization problems in heat transfer considered in this thesis was based on achieving a prescribed uniform temperature distribution on a specified subset of body boundary. In this case, the optimization problems were concerned with a boundary condition of the Dirichlet kind (temperature distribution). However, other aspects of the shape optimization in heat transfer need an extensive research. For example, minimizing the heat flux (a Neumann boundary condition) on a subset of body boundary is an important aspect of shape optimization in heat transfer, for example, when dealing with minimization of energy usage and carbon footprint, and promotion of sustainable processes.

6. A comprehensive comparison of the novel sensitivity analysis method presented in this thesis and the adjoint method for both heat transfer and aerodynamics parts is a main further research to be done.

7. In aerodynamics part, there are many steps in employed algorithm including grid generation, sensitivity analyses, the objective function definition, flow regime, optimization method. Thus it is impossible to deal with all of them in only one chapter and it demands further study. For example, the alternative methods for the airfoil parameterization such as B-spline and Non-uniform rational basis spline (NURBS), which with small

number of the points (design variables) produce a very smooth airfoil curve, require further study. Another further study includes alternative optimization methods.

8. The development of the proposed sensitivity analysis method for the aerodynamic shape optimization problems governed by Navier-Stokes equations is a main further research to be done. As the most of literature on the aerodynamic shape optimization is associated with the problems governed by these equations, such a development of the novel sensitivity analysis method for these equations is necessary.

8. References

[1] J.V. Beck, St. Clair C.R., and Blackwell B., *Inverse heat conduction* (1985).

[2] J.G. Stolz, Numerical Solutions to an Inverse Problem of Heat Conduction for Simple Shapes. *Journal of Heat Transfer*. **82** (1960) 20-25.

[3] T.J. Mirsepassi, *Heat-transfer Charts for Time-variable Boundary Conditions: Semi-infinite solid*. Aerojet-General Corporation (1958).

[4] T. Mirsepassi, Graphical evaluation of a convolution integral. *Mathematical Tables and Other Aids to Computation*. **13** (1959) 202-212.

[5] N. Shumakov, A method for the experimental study of the process of heating a solid body. *Soviet Physics-Technical Physics*. **2** (1957) 771-782.

[6] W.H. Giedt, The Determination of Transient Temperatures and Heat Transfer at a Gas-Metal Interface Applied to a 40-mm Gun Barrel. *Journal of Jet Propulsion*. **25** (1955) 158-162.

[7] J.V. Beck, Surface heat flux determination using an integral method. *Nuclear Engineering and Design*. **7** (1968) 170-178.

[8] C.H. Huang and Wang S.P., A three-dimensional inverse heat conduction problem in estimating surface heat flux by conjugate gradient method. *International Journal of Heat and Mass Transfer*. **42** (1999) 3387-3403.

[9] W.J. Minkowycz, Sparrow E.M., and Murthy J., *Handbook of numerical heat transfer*. J. Wiley (2006).

[10] E. Artyukhin, Reconstruction of the thermal conductivity coefficient from the solution of the nonlinear inverse problem. *Journal of Engineering Physics and Thermophysics*. **41** (1981) 1054-1058.

[11] O.M. Alifanov and Tryanin A.P., Determination of the coefficient of internal heat exchange and the effective thermal conductivity of a porous solid on the basis of a nonstationary experiment. *Journal of engineering physics*. **48** (1985) 356-365.

[12] L. Dantas and Orlande H., A function estimation approach for determining temperature-dependent thermophysical properties. *Inverse Problems in Engineering*. **3** (1996) 261-279.

[13] T. Jurkowski, Jarny Y., and Delaunay D., Estimation of thermal conductivity of thermoplastics under moulding conditions: an apparatus and an inverse algorithm. *International journal of heat and mass transfer*. **40** (1997) 4169-4181.

[14] C.-y. Yang, A linear inverse model for the temperature-dependent thermal conductivity determination in one-dimensional problems. *Applied Mathematical Modelling*. **22** (1998) 1-9.

[15] C.-y. Yang, Estimation of the temperature-dependent thermal conductivity in inverse heat conduction problems. *Applied Mathematical Modelling*. **23** (1999) 469-478.

[16] E. Divo, et al., Retrieval of multidimensional heat transfer coefficient distributions using an inverse BEM-based regularized algorithm: numerical and experimental results. *Engineering Analysis with Boundary Elements*. **29** (2005) 150-160.

[17] T.T.M. Onyango, Ingham D.B., and Lesnic D., Reconstruction of heat transfer coefficients using the boundary element method. *Computers & Mathematics with Applications*. **56** (2008) 114-126.

[18] M. Yamamoto and Zou J., Simultaneous reconstruction of the initial temperature and heat radiative coefficient. *Inverse problems*. **17** (2001) 1181.

[19] K. Masood, Messaoudi S., and Zaman F.D., Initial inverse problem in heat equation with Bessel operator. *International Journal of Heat and Mass Transfer*. **45** (2002) 2959-2965.
[20] F.-B. Liu, Particle Swarm Optimization-based algorithms for solving inverse heat conduction problems of estimating surface heat flux. *International Journal of Heat and Mass Transfer*. **55** (2012) 2062-2068.
[21] F.-B. Liu, Inverse estimation of wall heat flux by using particle swarm optimization algorithm with Gaussian mutation. *International Journal of Thermal Sciences*. **54** (2012) 62-69.
[22] D.T.W. Lin, et al., Inverse estimation of the unknown heat flux boundary with irregular shape fins. *International Journal of Heat and Mass Transfer*. **54** (2011) 5275-5285.
[23] C.K. Hsieh and Kassab A.J., A general method for the solution of inverse heat conduction problems with partially unknown system geometries. *International Journal of Heat and Mass Transfer*. **29** (1986) 47-58.
[24] G. Dulikravich and Martin T., Inverse shape and boundary condition problems and optimization in heat conduction. *Advances in Numerical Heat Transfer*. **1** (1996) 381-426.
[25] M. Özisjk and Orlande H., *Inverse heat transfer: fundamentals and applications*. Taylor & Francis (2000).
[26] A.N. Tikhonov and Arsenin V.I.A., *Solutions of ill-posed problems*. Winston (1977).
[27] O.M. Alifanov, *Inverse heat transfer problems*. Springer-Verlag (1994).
[28] J.V. Beck and Arnold K.J., *Parameter estimation in engineering and science*. Wiley (1977).
[29] J. Hadamard, *Lectures on Cauchy's Problem in Linear Partial Differential Equations*. Dover Publications (2003).
[30] A.N. Tikhonov, Inverse problems in heat conduction. *Journal of Engineering Physics and Thermophysics*. **29** (1975) 816-820.
[31] O.M.Alifanov, Determination of heat loads from a solution of the nonlinear inverse problem. *High temp*. **15** (1977) 498-504.
[32] O.M. Alifanov, Solution of an inverse problem of heat conduction by iteration methods. *Journal of Engineering Physics and Thermophysics*. **26** (1974) 471-476.
[33] Y. Bard, *Nonlinear parameter estimation*. Academic Press (1974).
[34] A. Tarantola, *Inverse problem theory and methods for model parameter estimation*. Society for Industrial and Applied Mathematics (2005).
[35] A. Bos, *Parameter estimation for scientists and engineers*. Wiley-Interscience (2007).
[36] R.C. Aster, Thurber C.H., and Borchers B., *Parameter estimation and inverse problems*. Elsevier Academic Press (2005).
[37] V.A. Morozov and Stessin M., *Regularization methods for ill-posed problems*. CRC press Boca Raton, FL: (1993).
[38] D. Calvetti, et al., Tikhonov regularization and the L-curve for large discrete ill-posed problems. *Journal of Computational and Applied Mathematics*. **123** (2000) 423-446.
[39] C.H. Huang and Ozisik M.N., Optimal regularization method to determine the strength of a plane surface heat source. *International Journal of Heat and Fluid Flow*. **12** (1991) 173-178.
[40] M. Özışık, *Heat conduction*. Wiley (1993).
[41] K. Kurpisz and Nowak A.J., *Inverse thermal problems*. Computational Mechanics Publications (1995).

[42] M.J. Colaço, Orlande H.R.B., and Dulikravich G.S., Inverse and optimization problems in heat transfer. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*. **28** (2006) 1-24.
[43] R.A. Meric, Shape design sensitivity analysis and optimization for nonlinear heat and electric conduction problems. *Numerical Heat Transfer, Part A Applications*. **34** (1998) 185-203.
[44] K. Dems and Mroz Z., Sensitivity analysis and optimal design of external boundaries and interfaces for heat conduction systems. *Journal of thermal stresses*. **21** (1998) 461-488.
[45] C.-H. Cheng and Wu C.-Y., An approach combining body-fitted grid generation and conjugate gradient methods for shape design in heat conduction problems. *Numerical Heat Transfer, Part B: Fundamentals: An International Journal of Computation and Methodology*. **37** (2000) 69 - 83.
[46] M. Özişik, *Finite difference methods in heat transfer*. CRC Press (1994).
[47] T.J. Chung, *Computational Fluid Dynamics*. Cambridge University Press (2010).
[48] Y. Çengel and Turner R., *Fundamentals of thermal-fluid sciences*. McGraw-Hill Companies (2004).
[49] J.P. Holman, *Heat transfer*. McGraw-Hill (2002).
[50] F.P. Incropera and DeWitt D.P., *Fundamentals of heat and mass transfer*. John Wiley (2007).
[51] A. Bejan, *Heat transfer*. John Wiley & Sons, Inc. (1993).
[52] Y.A. Çengel, *Heat transfer: a practical approach*. WBC McGraw-Hill (1998).
[53] C.P. Kothandaraman and Subramanyan S., *Fundamentals of heat and mass transfer*. New Age International (2008).
[54] C.-H. Cheng, Lin H.-H., and Aung W., Optimal shape design for packaging containing heating elements by inverse heat transfer method. *Heat and Mass Transfer*. **39** (2003) 687-692.
[55] C.-H. Cheng and Chang M.-H., Shape Identification by Inverse Heat Transfer Method. *Journal of Heat Transfer*. **125** (2003) 224-231.
[56] C.H. Lan, Cheng C.H., and Wu C.Y., SHAPE DESIGN FOR HEAT CONDUCTION PROBLEMS USING CURVILINEAR GRID GENERATION, CONJUGATE GRADIENT, AND REDISTRIBUTION METHODS. *Numerical Heat Transfer: Applications*. **39** (2001) 487-510.
[57] M. Yoon, Ha S.-H., and Cho S., Isogeometric shape design optimization of heat conduction problems. *International Journal of Heat and Mass Transfer*. **62** (2013) 272-285.
[58] T.J.R. Hughes, Cottrell J.A., and Bazilevs Y., Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*. **194** (2005) 4135-4195.
[59] S.M.H. Sarvari, Optimal Geometry Design of Radiative Enclosures Using the Genetic Algorithm. *Numerical Heat Transfer, Part A: Applications*. **52** (2007) 127-143.
[60] C. Zhuang, Xiong Z., and Ding H., Topology Optimization of the Transient Heat Conduction Problem on a Triangular Mesh. *Numerical Heat Transfer, Part B: Fundamentals*. **64** (2013) 239-262.
[61] C. Zhuang, Xiong Z., and Ding H., Topology optimization of multi-material for the heat conduction problem based on the level set method. *Engineering Optimization*. **42** (2010) 811-831.
[62] S.-H. Ahn and Cho S., Level Set–Based Topological Shape Optimization of Heat Conduction Problems Considering Design-Dependent Convection Boundary. *Numerical Heat Transfer, Part B: Fundamentals*. **58** (2010) 304-322.

[63] S.-H. Ha and Cho S., Topological Shape Optimization of Heat Conduction Problems using Level Set Approach. *Numerical Heat Transfer, Part B: Fundamentals*. **48** (2005) 67-88.
[64] S.-H. Ha and Cho S., Level Set-based Topological Shape Optimization of Nonlinear Heat Conduction Problems. *Numerical Heat Transfer, Part B: Fundamentals*. **54** (2008) 454-475.
[65] M.-G. Kim, Ha S.-H., and Cho S., Level Set-Based Topological Shape Optimization of Nonlinear Heat Conduction Problems Using Topological Derivatives #. *Mechanics Based Design of Structures and Machines*. **37** (2009) 550-582.
[66] M. Burger, A framework for the construction of level set methods for shape optimization and reconstruction. *Interfaces and Free Boundaries*. **5** (2003) 301-329.
[67] S. Osher and Fedkiw R., *Level Set Methods and Dynamic Implicit Surfaces*. Springer (2003).
[68] C.-H. Huang and Chaing M.-T., A transient three-dimensional inverse geometry problem in estimating the space and time-dependent irregular boundary shapes. *International Journal of Heat and Mass Transfer*. **51** (2008) 5238-5246.
[69] C.-H. Huang and Chen C.-A., A three-dimensional inverse geometry problem in estimating the space and time-dependent shape of an irregular internal cavity. *International Journal of Heat and Mass Transfer*. **52** (2009) 2079-2091.
[70] C.-H. Huang and Chaing M.-T., A three-dimensional inverse geometry problem in identifying irregular boundary configurations. *International Journal of Thermal Sciences*. **48** (2009) 502-513.
[71] C.-H. Huang and Chen Y.-H., An optimal design problem in determining non-uniform fin heights and widths for an impingement heat sink module. *Applied Thermal Engineering*. **63** (2014) 481-494.
[72] H. Cheng-Hung, Cheng-Tso W., and Hsi-Mei C., An inverse geometry problem in determining the interfacial surfaces of a three layers structure based on the desired system heat flux, in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, (2011) 348-352.
[73] C.-H. Huang and Wuchiu C.-T., Optimum Shape Design for a Three Layer Interfacial Surfaces. *Journal of Thermophysics and Heat Transfer*. **25** (2011) 476-480.
[74] J.F. Thompson, Thames F.C., and Mastin C.W., Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. *Journal of Computational Physics*. **15** (1974) 299-319.
[75] J. Thompson, Warsi Z., and Mastin C., *Numerical grid generation: foundations and applications*. North-Holland (1985).
[76] J. Thompson, Soni B., and Weatherill N., *Handbook of grid generation*. CRC Press (1999).
[77] P. Knupp and Steinberg S., *The fundamentals of grid generation*. CRC Press (1993).
[78] M. Farrashkhalvat and Miles J.P., *Basic structured grid generation with an introduction to unstructured grid generation*. Butterworth Heinemann (2003).
[79] O.C. Zienkiewicz, Taylor R.L., and Zhu J.Z., *The finite element method: its basis and fundamentals*. Elsevier Butterworth-Heinemann (2005).
[80] M. de Berg, et al., *Computational Geometry: Algorithms and Applications*. Springer (2008).
[81] F.P. Preparata and Shamos M.I., *Computational geometry: an introduction*. Springer-Verlag (1988).
[82] S.L. Devadoss and O'Rourke J., *Discrete and Computational Geometry*. Princeton University Press (2011).

[83] W.J. Gordon and Hall C.A., Construction of curvilinear co-ordinate systems and applications to mesh generation. *International Journal for Numerical Methods in Engineering*. **7** (1973) 461-477.
[84] P. Rattanadecho and Wongwises S., Simulation of freezing of water-saturated porous media in a rectangular cavity under multiple heat sources with different temperature using a combined transfinite interpolation and PDE methods. *Computers & Chemical Engineering*. **31** (2007) 318-333.
[85] C.W. Mastin, Error induced by coordinate systems. DTIC Document, (1982).
[86] Y. Jarny, Ozisik M.N., and Bardon J.P., A general optimization method using adjoint equation for solving multidimensional inverse heat conduction. *International Journal of Heat and Mass Transfer*. **34** (1991) 2911-2919.
[87] K. Levenberg, A method for the solution of certain nonlinear problems in least squares. *Quart. Appl. Math*. **2** (1944) 164–168.
[88] D.W. Marquardt, An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*. **11** (1963) 431-441.
[89] E. Chong and Żak S., *An introduction to optimization*. Wiley-Interscience (2008).
[90] J. Moré, The Levenberg-Marquardt algorithm: Implementation and theory, in *Numerical Analysis*, G.A. Watson (Ed.). Springer Berlin Heidelberg, (1978) 105-116.
[91] J.E. Dennis and Schnabel R.B., *Numerical methods for unconstrained optimization and nonlinear equations*. Society for Industrial and Applied Mathematics (1996).
[92] C.-H. Lan, Cheng C.-H., and Wu C.-Y., SHAPE DESIGN FOR HEAT CONDUCTION PROBLEMS USING CURVILINEAR GRID GENERATION, CONJUGATE GRADIENT, AND REDISTRIBUTION METHODS. *Numerical Heat Transfer, Part A: Applications: An International Journal of Computation and Methodology*. **39** (2001) 487 - 510.
[93] J. Bonnans, *Numerical optimization: theoretical and practical aspects*. Springer (2003).
[94] J. Nocedal and Wright S., *Numerical optimization*. Springer (1999).
[95] A. Antoniou and Lu W., *Practical optimization: algorithms and engineering applications*. Springer (2007).
[96] P.E. Gill, Murray W., and Wright M.H., *Practical optimization*. Academic Press (1981).
[97] R. Fletcher, *Practical methods of optimization*. Wiley (1987).
[98] S.S. Rao, *Engineering optimization: theory and practice*. Wiley (1996).
[99] R. Fletcher and Reeves C., Function minimization by conjugate gradients. *The computer journal*. **7** (1964) 149.
[100] E. Polak and Ribiere G., Note sur la convergence de méthodes de directions conjuguées. *Revue Française d'Informatique et de Recherche Opérationnelle*. **16** (1969) 35–43.
[101] J.W. Daniel, *The approximate minimization of functionals*. Prentice-Hall (1971).
[102] J. Blazek, *Computational Fluid Dynamics: Principles and Applications: (Book with accompanying CD)*. Elsevier Science & Technology (2005).
[103] R.M. Hicks, Murman E.M., and Vanderplaats G.N., An assessment of airfoil design by numerical optimization. (1974).
[104] R. Hicks and Henne P., WING DESIGN BY NUMERICAL OPTIMIZATION. *Journal of Aircraft*. **15** (1978).
[105] O. Pironneau, On optimum design in fluid mechanics. *Journal of Fluid Mechanics*. **64** (1974) 97-110.
[106] A. Jameson, Aerodynamic design via control theory. *J Sci Comput*. **3** (1988) 233-260.

[107] A. Jameson, Computational aerodynamics for aircraft design. *Science(Washington)*. **245** (1989) 361-371.
[108] A. Jameson, Optimum aerodynamic design using CFD and control theory. *AIAA paper*. **1729** (1995) 124-131.
[109] A. Jameson and Reuther J., *Control theory based airfoil design using the Euler equations*. Research Institute for Advanced Computer Science, NASA Ames Research Center (1994).
[110] M.B. Giles and Pierce N.A., Adjoint equations in CFD: duality, boundary conditions and solution behaviour. *AIAA paper*. **97** (1997) 1850.
[111] M.B. Giles and Pierce N.A., On the properties of solutions of the adjoint Euler equations. *Numerical Methods for Fluid Dynamics VI. ICFD*. (1998) 1-16.
[112] M.B. Giles, *Discrete adjoint approximations with shocks*. New York: Springer (2003).
[113] M.B. Giles, et al., Algorithm Developments for Discrete Adjoint Methods. *AIAA journal*. **41** (2003) 198-205.
[114] O. Baysal and Eleshaky M.E., Aerodynamic design optimization using sensitivity analysis and computational fluid dynamics. (1991).
[115] S. Ta'asan, Kuruvila G., and Salas M., Aerodynamic design and optimization in one shot, in *AIAA, Aerospace Sciences Meeting and Exhibit, 30 th, Reno, NV*, (1992) 1992.
[116] H. Cabuk, Sung C.-H., and Modi V., Adjoint operator approach to shape design for internal incompressible flows, in *Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences (ICIDES-3)*, (1991) 391-404.
[117] M. Desai and Ito K., Optimal controls of Navier-Stokes equations. *SIAM Journal on Control and Optimization*. **32** (1994) 1428-1446.
[118] J. Elliott and Peraire J., Aerodynamic design using unstructured meshes. *AIAA paper*. (1996).
[119] J. Elliott and Peraire J., Aerodynamic optimization on unstructured meshes with viscous effects. *AIAA paper*. (1997).
[120] J. Elliott and Peraire J., Practical 3D aerodynamic design and optimization using unstructured meshes. *AIAA J*. **35** (1997) 1479-1485.
[121] W.K. Anderson and Venkatakrishnan V., Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Computers & Fluids*. **28** (1999) 443-480.
[122] W.K. Anderson and Venkatakrishnan V., Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation. (1997).
[123] L. Gonzalez, et al., Multidisciplinary Aircraft Design and Optimisation Using a Robust Evolutionary Technique with Variable Fidelity Models. *AIAA paper*. **4625** (2004).
[124] I.C. Parmee and Watson A.H., Preliminary airframe design using co-evolutionary multiobjective genetic algorithms, in *Proceedings of the Genetic and Evolutionary Computation Conference*, (1999) 1657-1665.
[125] S. Obayashi, Multidisciplinary design optimization of aircraft wing planform based on evolutionary algorithms, in *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*. IEEE, (1998) 3148-3153.
[126] A. Oyama, Liou M.-S., and Obayashi S., Transonic axial-flow blade shape optimization using evolutionary algorithm and three-dimensional Navier-Stokes solver, in *9th AIAA/ISSMO Symposium and Exhibit on Multidisciplinary Analysis and Optimization, Atlanta, GA*, (2002).

[127] H.-S. Chung, Choi S., and Alonso J.J., Supersonic business jet design using knowledge-based genetic algorithm with adaptive, unstructured grid methodology, in *21st Applied Aerodynamics Conference*, (2003).
[128] A. Jameson and Ou K., Optimization Methods in Computational Fluid Dynamics, in *Encyclopedia of Aerospace Engineering*. John Wiley & Sons, Ltd, (2010).
[129] D. Thévenin and Janiga G., *Optimization and Computational Fluid Dynamics*. Springer Verlag (2008).
[130] B. Mohammadi and Pironneau O., *Applied Shape Optimization for Fluids*. Oxford University Press (2009).
[131] P. Castonguay and Nadarajah S., Effect of shape parameterization on aerodynamic shape optimization, in *45th AIAA Aerospace Sciences Meeting and Exhibit*, (2007) 8-11.
[132] J.D. Anderson, *Fundamentals of aerodynamics*. McGraw-Hill (2001).
[133] M.W. Kutta, Lifting forces in flowing fluids. (1902).
[134] E.L. Houghton, et al., *Aerodynamics for Engineering Students*. Elsevier Science (2012).
[135] M. Drela, *XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils* (1989).
[136] R.L. Fearn, Airfoil Aerodynamics Using Panel Methods. *The Mathematica Journal*. **10** (2008).
[137] XFLR5: An analysis tool for airfoils, wings and planes. http://www.xflr5.com/xflr5.htm.
[138] J.A. Samareh, A survey of shape parameterization techniques, in *NASA CONFERENCE PUBLICATION*. Citeseer, (1999) 333-344.
[139] H. Sobieczky, Parametric airfoils and wings. *Notes on Numerical Fluid Mechanics*. **68** (1999) 71-88.
[140] J.-A. Désidéri, Abou El Majd B., and Janka A., Nested and self-adaptive Bézier parameterizations for shape optimization. *Journal of Computational Physics*. **224** (2007) 117-131.
[141] K. Park, et al., Optimal Design of Airfoil with High Aspect Ratio in Unmanned Aerial Vehicles. *International Journal of Mechanical, Industrial, and Aerospace Engineering*. **2** (2008) 66-72.
[142] M.H. Sohn and Lee K.J., Bezier curve application in the shape optimization of transonic airfoils. *AIAA paper*. **4523** (2000) 2000.
[143] K. Park, et al., Aerodynamics and optimization of airfoil under ground effect. *International Journal of Mechanical Systems Science and Engineering*. **1** (2009).
[144] M. Kouhi, et al., Aerodynamic shape optimization using adaptive remeshing. (2010).
[145] R. Derksen and Rogalsky T., Bezier-PARSEC: An optimized aerofoil parameterization for design. *Advances in Engineering Software*. **41** (2010) 923-930.
[146] W.K. Anderson and Bonhaus D.L., Aerodynamic design on unstructured grids for turbulent flows, in *NASA TM*. Citeseer, (1997).
[147] T. Rogalsky, Kocabiyik S., and Derksen R., Differential evolution in aerodynamic optimization. *Canadian Aeronautics and Space Journal*. **46** (2000) 183-190.
[148] B.M. Kulfan and Bussoletti J.E., Fundamental parametric geometry representations for aircraft component shapes. *AIAA paper*. **6948** (2006) 2006.
[149] B.A. Gardner and Selig M.S., Airfoil design using a genetic algorithm and an inverse method. *AIAA paper*. **43** (2003).
[150] R.K.N. Parasaram and Charyulu T., AIRFOIL PROFILE DESIGN BY REVERSE ENGINEERING BEZIER CURVE. (2012).
[151] J.F. Coimbra, *Aero-Acoustic Optimization of Airfoils for Wind Turbines*. Master's thesis, Instituto Superior Técnico (2012).

[152] J. Fazil and Jayakumar V., Investigation of airfoil profile design using reverse engineering Bezier curve. *Journal of Engineering and Applied Sciences*. **6** (2006).
[153] F. Grasso, Design and Optimization of Tidal Turbine Airfoil. *Journal of Aircraft*. **49** (2012) 636-643.
[154] V. Prasanth and Lal S.A., Bezier Parameterization of an airfoil using genetic algorithm. (2010).
[155] M. Tanaka and Dulikravich G., AIRFOIL SHAPE DESIGN USING THE NAVIER-STOKES EQUATIONS. *Inverse Problems in Engineering Mechanics*. (1998) 409.
[156] D.F. Rogers, *An introduction to NURBS: with historical perspective*. Morgan Kaufmann Pub (2001).
[157] M. Hepperle, Javafoil—analysis of airfoils. *See* http://www.mh-aerotools.de/airfoils/javafoil.htm. (2008).
[158] C.G. Broyden, The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*. **6** (1970) 76-90.
[159] R. Fletcher, A new approach to variable metric algorithms. *The computer journal*. **13** (1970) 317-322.
[160] D. Goldfarb, Generating conjugate directions without line searches using factorized variable metric updating formulas. *Mathematical Programming*. **13** (1977) 94-110.
[161] D.F. Shanno, Conditioning of quasi-Newton methods for function minimization. *Mathematics of computation*. **24** (1970) 647-656.
[162] S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons (2009).
[163] A. Jameson, Aerodynamic shape optimization using the adjoint method. *Lectures at the Von Karman Institute, Brussels*. (2003).
[164] A. Jameson, Martinelli L., and Pierce N., Optimum aerodynamic design using the Navier–Stokes equations. *Theoretical and Computational Fluid Dynamics*. **10** (1998) 213-237.
[165] M.H. Straathof, *Shape Parameterization in Aircraft Design: A Novel Method, Based on B-splines* (2012).
[166] M.B. Giles and Pierce N.A., An introduction to the adjoint approach to design. *Flow, turbulence and combustion*. **65** (2000) 393-415.
[167] W.K. Anderson and Bonhaus D.L., Airfoil design on unstructured grids for turbulent flows. *AIAA journal*. **37** (1999) 185-191.
[168] T.D. Economon, Palacios F., and Alonso J.J., Optimal shape design for open rotor blades. *AIAA paper*. **3018** (2012) 2012.
[169] S.K. Nadarajah, *The discrete adjoint approach to aerodynamic shape optimization*. Citeseer (2003).